



Universidad
Carlos III de Madrid

Departamento de Informática

PROYECTO FIN DE GRADO

DISEÑO Y DESARROLLO DE UNA CARTERA INTELIGENTE PARA MEJORAR LAS ACTIVIDADES DE MICROPAGOS

Autor: Carlos Alcalá García

Tutor: Marco Romano

Leganés, Junio 2014

Título: Diseño y desarrollo de una cartera inteligente para mejorar las actividades de micropagos.

Autor: Carlos Alcalá García

Director: Marco Romano

EL TRIBUNAL

Presidente: _____

Vocal:

Secretario:

Realizado el acto de defensa y lectura del Proyecto Fin de Grado el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE





Resumen

El uso de los smartphones ha experimentado un gran crecimiento en los últimos años. Estos dispositivos cuentan con lectores NFC, cámaras, conexión a internet y una gran cantidad de sensores que permiten saber, por ejemplo, la ubicación o la temperatura. Esto facilita que el dispositivo se adapte al entorno y a los hábitos del usuario permitiendo el desarrollo de sistemas ubicuos que reduzcan el esfuerzo de los usuarios al realizar tareas cotidianas.

En la actualidad existen aplicaciones que hacen uso de la tecnología NFC y la geolocalización para realizar compras en los establecimientos a través del Smartphone. Muchos comercios y restaurantes se están adaptando a estas nuevas tecnologías ya que se prevé que en los próximos años una gran cantidad de pagos se realicen por medio de aplicaciones móviles.

Sin embargo, los mobile wallets actuales utilizan interacciones de pago distintas que pueden llevar a confusión a los usuarios y faltan guías para diseñar la interacción durante los pagos con dispositivos móviles.

Este proyecto se centra en la definición de dos modelos de interacción y en el desarrollo de un prototipo de aplicación móvil, con la finalidad de superar las limitaciones de las aplicaciones de pago disponibles en la actualidad y que permita demostrar la eficacia y la eficiencia de los modelos de pago a través del smartphone.

Se han desarrollado dos modelos para realizar los pagos: *cash desk*, donde la interacción es similar al uso de una tarjeta de crédito actual y *check-in*, en el que el usuario se registra al acceder al establecimiento, enviando su información sin necesidad de realizar ninguna tarea más al realizar el pago.

Palabras clave: Computación Ubicua, Smartphone, Mobile Wallet, modelos de interacción de pago, Cash Desk, Check In

Abstract

The use of the smartphones have experienced a faster growth in the recent years. This devices include NFC readers, cameras, internet connection, and a lot of sensors which allow knowing, e.g., the user's location or the temperature. This enables the device to adapt to the environment and "learn" about the user's profile, letting the developers to create ubiquitous systems that reduce the daily tasks' effort.

Nowadays there are some apps that use the NFC technology and geolocation to make payments in stores using the smartphone. A lot of restaurants and other establishments are getting adapted to these new payment methods due to in the next years is being expected that a substantial amount of the payments will be made using the smartphone's apps.

However, current mobile wallet apps use different interaction to perform the purchases, this may lead the users to get confused. There is a lack of guidelines to design the payment interaction.

This bachelor thesis focuses on the definition of two interaction models and the development of a mobile application prototype aiming to overcome the current applications' limitations and demonstrate the efficacy and efficiency of these models.

In this project two new mobile payment interaction models have been designed: cash desk, where the interaction is similar to paying with a credit card, and check-in, where the consumer just has to register when he/she enters in the store, not having to do any other task to make the payment.

Keywords: Ubiquitous computing, Smartphone, Mobile Wallet, payment interaction models, Cash Desk, Check In

Índice

1. Introducción	11
1.1 Contexto	12
1.2 Objetivo	13
1.3 Método de diseño e implementación	13
1.4 Estructura del documento.....	14
2. Estado del arte	16
2.1. Aplicaciones Mobile Wallet actuales	16
3. Modelos de interacción y análisis	24
3.1. Modelos de interacción.....	24
3.1.1 Cash Desk	24
3.1.2 Check In	24
3.2. Escenarios.....	25
3.2.1 Escenario 1: Tienda de deportes	25
3.2.2 Escenario 2: Cafetería.....	26
3.2.3 Escenario 3: Mercadillo	28
3.3. Análisis de requisitos.....	29
3.3.1 Requisitos funcionales.....	30
3.3.2 Requisitos del entorno	36
3.3.3 Requisitos de datos	37
3.3.4 Perfil de usuario	38
3.3.5 Requisitos de usabilidad.....	39
4. Diseño e implementación	41
4.1 Introducción y proceso de diseño	41
4.2 Tecnología	42
4.2.1 RFID	42
4.2.2 Códigos QR	45
4.3 Workflow: Diseño de la interacción	46
4.3.1 Cash Desk	46
4.3.2 Check In	48
4.3.3 Transferencias: Comparte los gastos	50
4.4. Diseño de la interfaz móvil.....	51
4.4.1 Registro	51



4.4.2 Acceso a la aplicación.....	52
4.4.3 Pantalla Principal. Cartera.....	53
4.4.4 Navegación.....	54
4.4.5 Formulario de nueva tarjeta	56
4.4.6 Sección Explora	58
4.4.7 Sección Actividad Reciente	59
4.4.8 Sección Cupones y descuentos	60
4.4.9 Sección Comparte los gastos.....	60
4.4.10 Sección Configuración	61
4.4.11 Proceso de pago	62
4.4.12 Recibo.....	64
4.5. Detalles de Implementación	64
4.5.1 Arquitectura del sistema	64
4.5.2 Base de datos	66
4.5.3 Frameworks utilizados	68
5. Evaluación	79
5.1 Test de evaluación de la usabilidad.....	79
5.1.1 Experimento	80
5.2 Consideraciones a la hora de diseñar un Mobile Wallet: Lecciones aprendidas	82
6. Gestión del proyecto	84
6.1 Ciclo de vida del desarrollo de software	84
6.2 Planificación	85
6.3 Presupuesto	86
6.3.1 Recursos humanos	87
6.3.2 Recursos materiales	88
6.3.3 Resumen de costes.....	89
7. Conclusiones y líneas futuras de desarrollo.....	90
7.1 Conclusiones.....	90
7.2 Líneas futuras de desarrollo.....	92
Bibliografía	93
Anexo I. Diseño de la base de datos.....	97
Anexo II. Encuesta de evaluación.....	100
Anexo III. Interfaz de mWallet.....	101
Anexo IV - Project Summary.....	107



Índice figuras

Figura 1. Design Science approach [35]	14
Figura 2. Proceso de diseño centrado en el usuario	41
Figura 3 Arquitectura de un sistema RFID.....	42
Figura 4 Código QR. [22].....	45
Figura 5 Partes del Código QR [22]	45
Figura 6. Flujo de trabajo de la modalidad de pago Cash Desk usando NFC	47
Figura 7. Flujo de trabajo de la modalidad de pago Cash Desk usando QR.....	48
Figura 8. Flujo de trabajo de la modalidad de pago Check In usando QR.....	48
Figura 9. Flujo de trabajo de la modalidad de pago Check In usando geolocalización.....	49
Figura 10. Flujo de trabajo de la modalidad de pago Check In usando NFC.....	49
Figura 11. Proceso de diseño de la pantalla de registro	52
Figura 12. Proceso de diseño de la pantalla de acceso a la aplicación	53
Figura 13. Proceso de diseño de la sección Cartera.....	53
Figura 14. Proceso de diseño del menú lateral de navegación.....	55
Figura 15. Proceso de diseño del formulario de nueva tarjeta.....	56
Figura 16. Selector de fecha de caducidad incluido en el diseño final	57
Figura 17. Icono de tarjeta para abrir el formulario “Nueva Tarjeta”	57
Figura 18. Proceso de diseño de la sección Explora.....	58
Figura 19. Marcador de un establecimiento cercano en la sección explora.....	58
Figura 20. Proceso de diseño de la sección Actividad Reciente.....	59
Figura 21. Proceso de diseño de la sección Cupones y Descuentos	60
Figura 22. Proceso de diseño de la sección Comparte los Gastos	61
Figura 23. Proceso de diseño de la sección Configuración	61
Figura 24. Proceso de diseño del formulario de check in	62
Figura 25. Barra indicadora de check in activo incluida en el diseño final de la aplicación.....	63
Figura 26. Proceso de diseño del pago a través de cash desk	63
Figura 27. Proceso de diseño del Recibo.....	64
Figura 28. Arquitectura de mWallet.....	65
Figura 29. Diseño relacional de la base de datos MySQL.....	66
Figura 30. Diagrama explicativo del funcionamiento de los fragments.....	68
Figura 31. Diagrama de flujo del protocolo de comunicación GCM	72
Figura 32. Ciclo de vida en cascada.....	84
Figura 33. Diagrama de Gantt de la planificación del proyecto	86
Figure 1. Design Science approach [35]	111

Índice de tablas

Tabla 1. Principales características de Google Wallet.....	17
Tabla 2. Principales características de PayPal.....	18
Tabla 3. Principales características de BBVA Wallet	19
Tabla 4. Principales características de Isis.....	20
Tabla 5. Principales características de Bemoov	21
Tabla 6. Principales características de PingPing.....	22
Tabla 7. Principales características de QuickTap.....	23
Tabla 8. Comparativa de las aplicaciones Mobile Wallet.....	23
Tabla 9. Comparativa entre las modalidades de pago del Escenario 1.....	26
Tabla 10. Comparativa entre las modalidades de pago del Escenario 2.....	27
Tabla 11. Comparativa entre las modalidades de pago del Escenario 3.....	28
Tabla 12. Requisito funcional 001	30
Tabla 13. Requisito funcional 002	30
Tabla 14. Requisito funcional 003	30
Tabla 15. Requisito funcional 004	30
Tabla 16. Requisito funcional 005	31
Tabla 17. Requisito funcional 006	31
Tabla 18. Requisito funcional 007	31
Tabla 19. Requisito funcional 008	31
Tabla 20. Requisito funcional 009	32
Tabla 21. Requisito funcional 010	32
Tabla 22. Requisito funcional 011	32
Tabla 23. Requisito funcional 012	32
Tabla 24. Requisito funcional 013	33
Tabla 25. Requisito funcional 014	33
Tabla 26. Requisito funcional 015	33
Tabla 27. Requisito funcional 016	33
Tabla 28. Requisito funcional 017	34
Tabla 29. Requisito funcional 018	34
Tabla 30. Requisito funcional 019	34
Tabla 31. Requisito funcional 020	34
Tabla 32. Requisito funcional 021	35
Tabla 33. Requisito funcional 022	35
Tabla 34. Requisito funcional 023	35



Tabla 35. Requisito funcional 024	35
Tabla 36. Requisito funcional 025	36
Tabla 37. Requisito de entorno 001	36
Tabla 38. Requisito de entorno 002	36
Tabla 39. Requisito de entorno 003	36
Tabla 40. Requisito de entorno 004	37
Tabla 41. Requisito de entorno 005	37
Tabla 42. Requisito de datos 001	37
Tabla 43. Requisito de datos 002	37
Tabla 44. Requisito de datos 003	38
Tabla 45. Requisito de datos 004	38
Tabla 46. Requisito de perfil de usuario 001.....	38
Tabla 47. Requisito de perfil de usuario 002.....	38
Tabla 48. Requisito de perfil de usuario 003.....	39
Tabla 49. Requisito de perfil de usuario 004.....	39
Tabla 50. Requisito de usabilidad 001.....	39
Tabla 51. Requisito de usabilidad 002.....	39
Tabla 52. Requisito de usabilidad 003.....	40
Tabla 53. Requisito usabilidad 004.....	40
Tabla 54. Requisito usabilidad 005.....	40
Tabla 55. Requisito usabilidad 006.....	40
Tabla 56. Ventajas y desventajas de las etiquetas pasivas	43
Tabla 57. Ventajas y desventajas de las etiquetas activas	43
Tabla 58. Modos de operación NFC	44
Tabla 59. Resultados	80
Tabla 60. Valoración media de mWallet	81
Tabla 61. Preferencias en la utilización de los distintos medios de pago	81
Tabla 62. Planificación del proyecto	85
Tabla 63. Costes de personal	88
Tabla 64. Costes materiales	88
Tabla 65. Resumen de costes	89

1. Introducción

Los dispositivos móviles de hoy en día están dotados de tecnologías que permiten a los usuarios estar conectados e informados permanentemente y en tiempo real. Aprovechando estas capacidades, se pueden desarrollar entornos ubicuos, que permiten a los usuarios realizar las tareas más cotidianas sin que sean conscientes de estar usando un dispositivo móvil, lo que les permite centrarse en la tarea.

Este trabajo ha sido realizado en el ámbito de la *Human Computer Interaction* (HCI) y en particular de la computación ubicua.

El 4 de junio de 2014, la web “Merchant Warehouse”, [1] [2] haciéndose eco de una encuesta de Carlisle & Gallagher Consulting group sobre la utilización que en un futuro se hará de los smartphones para realizar pagos y transacciones, predice que para el año 2017 la mitad de los usuarios de Smartphone de hoy en día utilizarán *mobile wallets* como su método de pago preferido.

Durante el año 2013, Starbucks procesó más de mil millones de dólares mediante pagos realizados a través del dispositivo móvil [3].

En España [4], la adopción de la tecnología *contactless* NFC está creciendo rápidamente y ya se realizan más de 1,7 millones de transacciones al mes haciendo uso de la misma. Entidades financieras como BBVA, Bankia o ING Direct están colaborando activamente con Visa en este proceso de adopción de los pagos a través del Smartphone.

¿Qué es un mobile wallet?

Un mobile wallet es la versión electrónica de la cartera que todos llevamos en nuestros bolsillos, con el añadido de poder beneficiarnos de todas las ventajas que este nuevo medio nos aporta.

La mayoría nos permiten llevar todas nuestras tarjetas de crédito en el dispositivo móvil, aplicar los cupones de descuento de los que disponemos, conocer la localización de establecimientos cercanos... Además se podrán gestionar todas las transacciones que hayamos realizado y ver los recibos de cada una de ellas.

Los mobile wallets deben proporcionar una herramienta de pago eficaz y fácil de utilizar.

1.1 Contexto

Pero, si este tipo de aplicaciones proporcionan tantas ventajas, ¿por qué no se ha impuesto aún este modo de pago?

Como se menciona en el artículo “Research advances for the mobile payments arena” [5], nos encontramos actualmente en un momento en que los dispositivos móviles cada vez tienen una capacidad de cómputo mayor y son más sofisticados. Estas características permiten que se puedan realizar pagos en “tiempo real” de manera efectiva y eficiente.

Sin embargo, las soluciones mobile wallets existentes en la actualidad se encuentran limitadas a una zona geográfica concreta y/o para operadores específicos, lo que hace que éstas disten mucho de que puedan ser utilizadas de una forma universal.

Además, estas aplicaciones suelen tener una curva de aprendizaje importante y no están pensadas para un público general.

En [5] los autores especifican que no existe una colaboración efectiva entre bancos y entidades financieras y los desarrolladores de mobile wallets, lo que es un requisito imprescindible para su funcionamiento.

Durante los últimos años han surgido una gran cantidad de mobile wallets y algunos de ellos han pasado del nivel de prototipo al mercado real. Desafortunadamente, muchas de ellas no han conseguido mantenerse activas.

En la Unión Europea [6] se han realizado esfuerzos para tratar de obtener una estandarización de los pagos a través de dispositivos móviles, de tal forma que sea posible dar un soporte a las aplicaciones que están apareciendo.

Para ello, es necesario diseñar y desarrollar nuevos servicios de pagos y ajustar el marco regulatorio y las leyes para esta nueva modalidad de pago.

Un ejemplo de esto es SEMOPS (Secure Mobile Payment Service), que tiene como objetivo desarrollar un sistema universal y estandarizado capaz de gestionar los pagos tanto nacionales como internacionales.

SEMOPS se centra en la seguridad, la privacidad, la confianza de los usuarios y la flexibilidad del sistema.

Existen algunos trabajos [7] que realizan estudios para descubrir las características, tanto funcionales como de interfaz, que debe cumplir todo mobile wallet.

Tras entrevistar a numerosos usuarios de distintos perfiles, el resultado obtenido sería una aplicación con muchas funcionalidades, lo que sería muy difícil de abarcar y podría provocar que los usuarios no la utilizaran por su dificultad.

Es necesario hacer nuevos estudios más empíricos y pruebas en entornos reales para descubrir que es lo que los usuarios buscan realmente.

Contexto socio-económico

El proyecto que se presenta se enmarca en un contexto socio-económico que, si bien está marcado por una fuerte crisis económica, a los efectos de mWallet se puede considerar positivo.

Una de las razones de ello es el gran desarrollo de las telecomunicaciones y la generalización del uso de dispositivos móviles. Esto hace que el mercado de las “apps” esté en pleno auge.

Además, el objetivo principal de esta aplicación es implementar en ella modelos de interacción en los pagos con móvil que puedan ser considerados útiles para la mayoría de usuarios, comercios y entidades financieras. Esta “estandarización” podría permitir la generalización de este modo de pago con smartphone, lo que a la larga podría ser un elemento dinamizador de la economía.

1.2 Objetivo

En el presente trabajo, se trata diseñar y desarrollar una cartera inteligente capaz de mejorar las actividades de micropagos.

Se trata de superar las limitaciones de los sistemas actuales debidas a la heterogeneidad que presentan las interfaces de los distintos mobile wallets que existen en la actualidad, utilizando en una misma aplicación la mayoría de las tecnologías existentes de tal modo que se amplíe el abanico de dispositivos que puedan utilizarla así como el número de usuarios potenciales.

Para conseguir esto, el objetivo primordial es desarrollar modalidades de interacción que permitan realizar los pagos sintetizando los modelos más comunes de las aplicaciones actuales.

Para ello, se ha diseñado, implementado y evaluado un prototipo basado en los modelos de interacción utilizados para comprobar su eficacia y eficiencia.

1.3 Método de diseño e implementación

La metodología de trabajo utilizada se corresponde con el proceso de diseño “Design Science approach” [8].

Esta metodología sigue un proceso iterativo que permite evaluar y diseñar artefactos de IT (tecnologías de la información), centrándose en la utilidad de estos y la facilidad de entenderlos. Al final del proceso se obtienen resultados de los que se puede aprender.

Se identifican 3 ciclos/fases dentro de este proceso de diseño:

- *Relevance Cycle*: En base al entorno actual, teniendo en cuenta diversos componentes (personas, sistema organizativo, sistema técnico, problemas y oportunidades) se obtienen unos requisitos en los que se basará la aplicación.
- *Rigor Cycle*: Se utilizan conocimientos previamente existentes de aplicaciones y estudios similares, aprovechando la experiencia de los mismos a la hora de diseñar nuestro prototipo. Por ejemplo, para el proceso de diseño se ha seguido un modelo centrado en

el usuario; para mostrar la información de las tarjetas se sigue la norma del Payment Card Industry Data Security Standard [34] (PCI DSS) entre otros.

- **Design Cycle:** En base a los requisitos y a los conocimientos y experiencias similares que hemos recopilado en las dos fases anteriores se construye y se diseña el prototipo. Por último, se realizará una evaluación con usuarios de donde se podrá obtener un *feedback*.

Este es un proceso iterativo que se repite de forma continua. El entorno va cambiando, aparecen nuevos requisitos y nuevos conocimientos relacionados y todo ello se reflejará de nuevo en el diseño.

La figura 1, representa los ciclos del proceso Design Science Research.

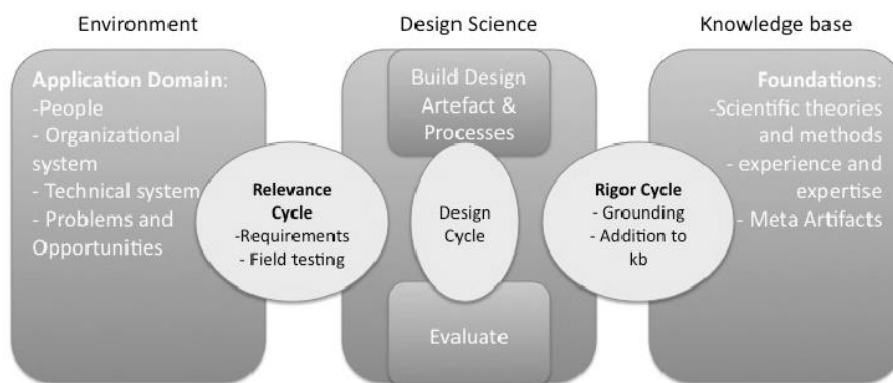


Figura 1. Design Science approach [35]

1.4 Estructura del documento

El documento se estructura en 7 capítulos, un apartado de bibliografía y cuatro anexos en los que se detallara el proceso seguido para desarrollar el proyecto.

El capítulo actual (*Capítulo 1 - Introducción*) muestra el panorama actual de los mobile wallets, poniéndose de manifiesto, que a pesar de que proporcionan muchas ventajas y que existen previsiones de futuro que auguran un uso importante, aún es necesario seguir trabajando para encontrar un estándar que permita que esta nueva modalidad de pago se pueda realizar de manera universal. Posteriormente, se detallan los objetivos que se persiguen con este proyecto y el proceso de desarrollo seguido para conseguirlos.

El *Capítulo 2 - Estado del arte*, muestra las aplicaciones mobile wallets más utilizadas en la actualidad. Aquí se analizan y se comparan las opciones existentes, permitiendo aprovechar el *user experience* de las mismas y conocer sus limitaciones.

En el *Capítulo 3 - Modelos de interacción y análisis*, se analizarán y formalizarán los modelos interactivos de pago desarrollados y que implementa mWallet. Se analizan distintos escenarios del mundo real donde se comparan ambas modalidades y se ponen de manifiesto las ventajas e inconvenientes en cada uno de ellos. A partir de estos escenarios se obtienen los requisitos que debe cumplir la aplicación y que son especificados al final de este capítulo.

El *Capítulo 4 - Diseño e implementación* detalla el proceso seguido para diseñar la aplicación móvil (*wireframe* y *mockup*) y el *workflow* de la misma. Además, se realiza un estudio de las tecnologías que se utilizan para implementar las dos modalidades de pago. Finalmente se detallan la arquitectura del sistema y los aspectos más relevantes de la implementación de mWallet.

En el *Capítulo 5 - Evaluación* se muestran los resultados obtenidos en una prueba piloto con usuarios que persigue comprobar el grado de usabilidad del diseño y la eficacia y eficiencia de los modelos de pago desarrollados.

El *Capítulo 6 - Gestión de Proyecto* describe las distintas fases del proyecto y la planificación realizada para llevar a cabo cada una de ellas. Con ello se obtendrá un presupuesto del proyecto, teniendo en cuenta los recursos humanos y materiales necesarios para llevarlo a cabo.

En el *Capítulo 7 - Conclusiones*, se resume el proyecto realizado y se muestran las líneas de futuro que se podrían seguir a partir del mismo.

El trabajo se completa con tres anexos:

- En el Anexo I - Diseño de la base de datos, se detallan las diferentes tablas y las relaciones existentes entre ellas.
- En el Anexo II - Encuesta de evaluación, se muestra la plantilla del formulario de evaluación que ha sido utilizada en el test piloto con usuarios.
- En el Anexo III - Interfaz de mWallet, se muestran capturas de pantalla de la aplicación desarrollada
- En el Anexo IV - Project summary, se cumple con el requisito de traducir una parte de la memoria al idioma inglés.

2. Estado del arte

En este capítulo se analizarán las principales aplicaciones que existen en la actualidad y que permiten realizar los pagos a través del Smartphone, así como las tendencias que siguen estas metodologías de pago.

2.1. Aplicaciones Mobile Wallet actuales

Las aplicaciones Mobile wallet permiten a los usuarios realizar transacciones a través de sus dispositivos móviles. Estas operaciones pueden ser tanto pagos por compras realizadas en establecimientos como transferencias entre cuentas.

Existen cuatro métodos principales de realizar pagos a través del dispositivo móvil [9]:

- A través de mensajes SMS. Al enviar el SMS el dispositivo se conecta con la red para realizar el pago.
- A través de páginas web. Los smartphones cuentan con conectividad a internet de modo que el usuario puede conectarse a una web que le permita realizar la transacción.
- A través de NFC o interacción “contactless”. Muchos dispositivos actuales cuentan con módulos NFC integrados. Para realizar el pago, el usuario acerca el dispositivo al lector y se transmite la información a través de radio frecuencias.
- Mediante códigos QR. El código se encuentra en el establecimiento o en un catálogo de productos y el usuario, para realizar la transacción, debe realizar una fotografía del código.

El objetivo de las aplicaciones Mobile Wallet es permitir al usuario llevar, en una aplicación de su dispositivo móvil, todas sus tarjetas de crédito y realizar pagos fácilmente a través de la misma. Además, este tipo de aplicaciones, en muchas ocasiones, también permiten a los usuarios utilizar las ofertas de los establecimientos, consultar sus operaciones recientes y gestionar de forma sencilla todas las tarjetas.

En Estados Unidos esta modalidad de pago está experimentando un gran crecimiento en los últimos años. Por ejemplo, Starbucks [3] generó más de mil millones de dólares en el año 2013 a través de pagos realizados con el dispositivo móvil.

Se prevé que el uso del NFC para este tipo de transacciones crezca en un 38% entre 2011 y 2016 [1].

Debido a este crecimiento muchos establecimientos se están adaptando para aceptar esta modalidad de pago.

En Europa, pese al crecimiento en el uso de los smartphones, este tipo de aplicaciones no está muy extendido. Ahora bien, en los últimos meses están apareciendo algunas aplicaciones que incorporan estos métodos de pago. Por ejemplo, BBVA ha lanzado BBVA Wallet [10], una aplicación disponible tanto para iOS como Android, que permite realizar transacciones con tarjetas tanto de forma online como en los propios establecimientos.

A continuación, se van a estudiar las aplicaciones actuales más relevantes y a comparar sus características y modalidades de pago.

Google Wallet

Es una de las aplicaciones que más popularidad está teniendo entre los usuarios. Requiere que el dispositivo tenga un módulo NFC integrado.

Para realizar el pago los usuarios introducen un PIN de seguridad para acceder a su lista de tarjetas, acercan el terminal al lector NFC que se encuentra en el comercio y automáticamente se realiza la transacción.

La tabla 1 resume las características principales de Google Wallet [11].

Google Wallet
<u>Tecnología necesaria para su uso:</u> <ul style="list-style-type: none"> • NFC • Conexión a internet
<u>Tipos de tarjeta soportados:</u> <ul style="list-style-type: none"> • Saldo de google Wallet (tarjeta prepago de google) • Soporte multitarjeta (Visa, Mastercard, American Express y Discover)
Permite realizar transferencias entre cuentas: <ul style="list-style-type: none"> • A través de la aplicación móvil • A través de gmail <p>Se puede realizar la transferencia desde el saldo de Google Wallet a otra cuenta personal o de otro usuario de Google Wallet</p>
Utiliza la geolocalización para indicar establecimientos cercanos que aceptan el pago a través de Google Wallet
<u>Seguridad y acceso:</u> <ul style="list-style-type: none"> • El acceso a la aplicación se realiza a través de un PIN de 4 dígitos • Permite deshabilitar el uso de Google Wallet desde la web en caso de pérdida del dispositivo móvil
<u>Otras características:</u> <ul style="list-style-type: none"> • Permite realizar pagos online a través de una página web • Sistema de notificaciones y envío de recibos al móvil después de realizar las compras • Permite gestionar ofertas y descuentos • Permite añadir tarjetas de fidelidad • Permite ver todos los movimientos realizados con la aplicación

Tabla 1. Principales características de Google Wallet

PayPal

La aplicación de PayPal [12] utiliza la cuenta de usuario de la web para enlazar los datos con la aplicación móvil.

En esta aplicación no se hace uso del NFC sino que se envían al establecimiento los datos con los que se va a realizar el pago. Cuando se realiza la compra el usuario solo debe indicar al vendedor su nombre y el pago se realizará automáticamente.

La tabla 2 resume las características principales de la aplicación de PayPal.

PayPal
<p><u>Tecnología necesaria para su uso:</u></p> <ul style="list-style-type: none"> • Conexión a internet • Geolocalización activada
<p><u>Tipos de tarjeta soportados:</u></p> <ul style="list-style-type: none"> • Soporte multitarjeta (Maestro, Visa, Mastercard, American Express y Discover) <p>Al igual que en la versión web, PayPal actúa como intermediario entre el establecimiento y la cuenta bancaria.</p>
<p>Permite realizar transferencias entre cuentas a través de la aplicación móvil utilizando el email de otra cuenta de PayPal o un número de teléfono.</p>
<p>Utiliza la geolocalización para indicar establecimientos cercanos que aceptan el pago. Además, la geolocalización es necesaria para poder realizar el registro en el establecimiento</p>
<p><u>Seguridad y acceso:</u></p> <ul style="list-style-type: none"> • El acceso a la aplicación se realiza a través de un PIN de 4 a 8 dígitos o a través del nombre de usuario y contraseña de la cuenta de PayPal
<p><u>Otras características:</u></p> <ul style="list-style-type: none"> • Permite realizar pedidos antes de llegar al establecimiento (pre-order) • Permite añadir tarjetas de crédito a través de la cámara del dispositivo móvil • Permite gestionar ofertas y descuentos • Permite ver todos los movimientos realizados con la aplicación

Tabla 2. Principales características de PayPal

BBVA Wallet

La aplicación BBVA Wallet [13] recientemente lanzada en España permite realizar pagos mediante un dispositivo móvil con NFC.

El usuario crea una tarjeta específica (Tarjeta wallet BBVA) para la aplicación que puede recargarse a través del saldo de otras cuentas.

La interacción para realizar las compras es similar a la realizada con Google Wallet. Se acerca el dispositivo al lector NFC y se realiza el pago. En este caso es necesario también que el usuario introduzca el PIN de seguridad en compras superiores a 20€ para que la transacción quede completada.

La tabla 3 resume las características principales de BBVA Wallet.

BBVA Wallet
<u>Tecnología necesaria para su uso:</u> <ul style="list-style-type: none"> • Conexión a internet • NFC, utilizando un adhesivo específico que se fija en la parte posterior del dispositivo
<u>Tipos de tarjeta soportados:</u> <ul style="list-style-type: none"> • Tarjeta wallet BBVA • Tarjetas de crédito
No permite realizar transferencias entre usuarios
Utiliza la geolocalización para indicar el establecimiento donde se ha realizado una compra pero no indica los establecimientos que aceptan el pago a través de la aplicación.
<u>Seguridad y acceso:</u> <ul style="list-style-type: none"> • El acceso se realiza a través del usuario y contraseña de la cuenta de BBVA • Es necesario indicar el PIN de seguridad en compras superiores a los 20€
<u>Otras características:</u> <ul style="list-style-type: none"> • Permite ver todos los movimientos realizados con la aplicación

Tabla 3. Principales características de BBVA Wallet

Isis

Esta aplicación se encuentra disponible para las operadoras: AT&T Mobility, T-Mobile USA y Verizon, tanto para el sistema operativo Android como para iOS.

Isis [14] permite realizar los pagos en los establecimientos haciendo uso de la tecnología NFC, al igual que Google Wallet o BBVA Wallet. El usuario, una vez seleccionada la tarjeta, solo debe acercar el dispositivo al lector NFC para realizar la compra.

La aplicación también permite al usuario gestionar las ofertas y añadir tarjetas de fidelidad.

La tabla 4 resume las características principales de la aplicación de Isis:

Isis
<p><u>Tecnología necesaria para su uso:</u></p> <ul style="list-style-type: none"> • Conexión a internet • NFC. En el caso de utilizar un dispositivo móvil con iOS es necesario utilizar una carcasa específica
<p><u>Tipos de tarjeta soportados:</u></p> <ul style="list-style-type: none"> • Soporte multitarjeta (American Express, Chase, Wells Fargo)
No permite realizar transferencias entre usuarios
Utiliza la geolocalización para indicar establecimientos cercanos que permiten el pago con Isis
<p><u>Seguridad y acceso:</u></p> <ul style="list-style-type: none"> • El acceso a la aplicación se realiza a través de un PIN de 4 dígitos. • Permite deshabilitar el uso de la aplicación desde la web en caso de pérdida o robo del dispositivo móvil. • Tarjeta SIM diseñada específicamente para dotar de mayor seguridad a los pagos.
<p><u>Otras características:</u></p> <ul style="list-style-type: none"> • Permite gestionar ofertas y descuentos • Permite añadir tarjetas de fidelidad

Tabla 4. Principales características de Isis

Bemoov

Bemoov [15] proporciona una serie de aplicaciones, cada una de ellas especializada en un tipo de compra:

- Pago por estacionamiento
- Comprar pases de esquí
- Entradas para eventos: teatro, cine o conciertos
- Billetes de transporte público
- Realizar pedidos de un catálogo de productos.

Estas aplicaciones utilizan distintas formas de interacción. Permite realizar los pagos mediante SMS, lectura de códigos QR o tecnología “contactless”.

Estos modos de pago, sobre todo el de SMS, permiten que el dispositivo móvil no deba disponer de la última tecnología, y por tanto, pueden ser usadas por una mayor cantidad de usuarios.

La tabla 5 resume las características principales de la aplicación de Bemoov:

Bemoov
<u>Tecnología necesaria para su uso:</u> <ul style="list-style-type: none"> • SMS • Cámara de fotos para poder realizar la lectura de códigos QR • Tecnología NFC
<u>Tipos de tarjeta soportados:</u> <ul style="list-style-type: none"> • Cuentas bancarias asociadas a CartaSì Eura, Banca Carige Banca Mediolanum • Soporte multitarjeta (Visa, Mastercard, American Express)
No permite realizar transferencias entre usuarios
No utiliza geolocalización
<u>Seguridad y acceso:</u> <ul style="list-style-type: none"> • Se debe indicar el CVC de la tarjeta de crédito en compras superiores a 10€ • Se recibe un SMS tras realizar cada pago

Tabla 5. Principales características de Bemoov

PingPing

Esta aplicación [16] se encuentra disponible para los usuarios de smartphones residentes en Bélgica. PingPing permite realizar los pagos a través de un mensaje SMS a un número de teléfono que se encuentra en el propio establecimiento.

Además de SMS, también se pueden realizar los pagos mediante contactless. Para ello es necesario colocar en la parte posterior del dispositivo una etiqueta. En esta modalidad de pago la interacción es similar a la que usa Google Wallet, BBVA Wallet o Isis.

La tabla 6 resume las características principales de la aplicación de PingPing:

PingPing
<u>Tecnología necesaria para su uso:</u> <ul style="list-style-type: none"> • SMS • Etiqueta NFC
<u>Tipos de tarjeta soportados:</u> <ul style="list-style-type: none"> • Utiliza un crédito propio de la aplicación, que puede ser recargado.
No permite realizar transferencias entre usuarios
No utiliza geolocalización
<u>Seguridad y acceso:</u> <ul style="list-style-type: none"> • Se utiliza los datos de la cuenta de PingPing y un código PIN

Tabla 6. Principales características de PingPing

QuickTap

Es una aplicación [17] diseñada específicamente para realizar pagos en máquinas expendedoras. El usuario puede pagar o bien usando una tarjeta prepago o bien usando su Smartphone.

Mediante la aplicación móvil el usuario solamente deberá acercar el dispositivo al lector NFC o, en caso de no disponer de un módulo NFC, escanear un código QR que se encuentra en la propia máquina.

La tabla 7 resume las características principales de la aplicación de QuickTap.

QuickTap
<u>Tecnología necesaria para su uso:</u> <ul style="list-style-type: none"> NFC Cámara de fotos
<u>Tipos de tarjeta soportados:</u> <ul style="list-style-type: none"> Utiliza un crédito propio de la aplicación, que puede ser recargado.
No permite realizar transferencias entre usuarios
<u>Otras características:</u> Permite utilizar promociones y descuentos

Tabla 7. Principales características de QuickTap

En la tabla 8 se puede ver una comparativa de las aplicaciones analizadas anteriormente.

	Internet	NFC	Geolocalización	Soporte Multitarjeta	Transferencias	PIN	Ofertas y descuentos	Versión web	SMS	QR
Google Wallet	X	X	X	X	X	X	X	X		
PayPal	X		X	X	X	X	X	X		X
BBVA Wallet	X	X		X		X		X		
Isis	X	X	X	X		X	X	X		
Bemoov		X		X					X	X
PingPing		X				X		X	X	
QuickTap	X	X					X			

Tabla 8. Comparativa de las aplicaciones Mobile Wallet

En este apartado se han analizado algunas de las aplicaciones actuales que permiten realizar transacciones a través de los smartphones. Sin embargo, existen otras muchas que utilizan formas de pago similares como Cityzi, que utiliza la tecnología NFC; o Payforit, que es utilizada para pagos a través de internet.

La mayoría de las aplicaciones mobile wallet comparten modelos de interacción similares, como el uso de la tecnología NFC o un PIN de acceso para otorgar seguridad a la aplicación en el caso de pérdida o robo del dispositivo.

3. Modelos de interacción y análisis

3.1. Modelos de interacción

En el capítulo 2 se han estudiado algunas de las aplicaciones que existen en la actualidad para realizar pagos a través de un dispositivo móvil. Éstas utilizan NFC para hacer los pagos o geolocalización para conocer establecimientos cercanos e incluso enviar la información del usuario a dicho establecimiento antes de hacer la compra.

Tomando como referencia los distintos métodos de pago que ofrecen estas aplicaciones y teniendo en cuenta sus limitaciones, se van a diseñar dos modelos de interacción que podrán ser utilizados como base a la hora de desarrollar nuevos mobile wallets.

En el proyecto se va a implementar un prototipo de aplicación que permita realizar las transacciones a través del Smartphone. Los dos modelos de interacción que se han desarrollado, Cash Desk y Check In, se basan en la *experiencia de usuario* con las aplicaciones ya existentes.

Al final de este capítulo se obtendrán los requisitos de un prototipo de aplicación que hará uso de estos modelos.

3.1.1 Cash Desk

El primer modelo de interacción desarrollado se denomina Cash Desk.

Con esta modalidad, se pretende trasladar al dispositivo móvil el proceso de pago que se realiza diariamente en las transacciones comerciales más habituales.

Se trata de realizar una compra o pagar un servicio de manera similar a como se hace habitualmente, con la diferencia de que se utiliza para ello un Smartphone.

El usuario lleva en su terminal todas las tarjetas de crédito que desee utilizar y en el momento de realizar el pago solo tiene que seleccionar la que prefiera y completar la transacción en la caja del establecimiento. El pago se llevará a cabo utilizando alguna de las tecnologías disponibles (NFC, Códigos QR...).

3.1.2 Check In

La segunda modalidad de pago desarrollada se denomina Check In.

El usuario se registra en un establecimiento enviando sus datos personales así como la información de la tarjeta que va a usar. Dicha información quedará almacenada en el servidor que utilice el establecimiento.

Posteriormente, si el usuario realiza una compra, bastará con que se identifique ante el personal del establecimiento y se completará el pago sin necesidad de ninguna otra acción por su parte.

En caso de que no realice ninguna compra, el usuario tendrá la posibilidad de cancelar el Check In, eliminándose todos los datos que facilitó al establecimiento.

El modo de registrarse un usuario en un establecimiento puede utilizar distintas tecnologías (Geolocalización, QR, NFC...).

Ofrece las mismas ventajas que la modalidad Cash Desk y además, en determinados escenarios, proporcionará una mayor agilidad y comodidad.

3.2. Escenarios

En este apartado se describirán varios escenarios que han sido diseñados en colaboración con usuarios potenciales de la aplicación móvil y en los que se utilizan los distintos métodos de pago que se pueden realizar con mWallet.

Para cada uno de los escenarios se compararán dos modelos y se estudiarán cuáles son las ventajas y desventajas de cada uno de estos métodos según el entorno en que se realice.

3.2.1 Escenario 1: Tienda de deportes

María ha decidido practicar running y va a una famosa tienda de deportes para adquirir unas zapatillas deportivas.

Es usuaria de la aplicación mWallet, que le permite utilizar varias modalidades de pago.

Cash Desk mediante NFC

María elige sus zapatillas, se dirige a la caja del establecimiento e indica que pagará con su teléfono móvil.

Selecciona en la aplicación mWallet su tarjeta de crédito y aproxima el dispositivo a un terminal del establecimiento equipado con un lector NFC.

Ambos dispositivos se ponen en contacto y se efectúa el pago.

En el teléfono de María aparece un recibo con todos los datos de la compra que acaba de realizar.

Check In mediante NFC

La tienda de deportes tiene habilitados varios puntos que cuentan con lectores NFC para realizar Check In.

Cuando María entra en el establecimiento abre mWallet, selecciona una tarjeta de crédito de su lista, acerca su teléfono móvil a uno de esos lectores NFC y realiza el Check In.

Tras elegir sus zapatillas deportivas decide pagarlas. Para ello y sin necesidad de sacar el móvil de su bolso, solo tiene que decir su nombre en la caja del establecimiento y se realizará el pago. En el teléfono de María aparecerá un recibo con los detalles de la operación.

La tabla 9, que se muestra a continuación, indica las principales ventajas e inconvenientes de las dos modalidades de pago reflejadas en este escenario.

Cash Desk	Check In
<ul style="list-style-type: none"> + Su similitud con el pago tradicional con tarjeta de crédito hace que el usuario se sienta más cómodo. + Mayor percepción de seguridad. 	<ul style="list-style-type: none"> + Más rápido en el momento del pago. El usuario no tiene que sacar el dispositivo móvil. + Mayor comodidad. El usuario puede olvidarse del móvil y concentrarse en sus compras una vez realizado el Check In. - El tiempo transcurrido desde que el cliente hace Check In hasta que realiza el pago puede provocar desconfianza en el usuario, especialmente si es un establecimiento en el que compra por primera vez.

Tabla 9. Comparativa entre las modalidades de pago del Escenario 1

3.2.2 Escenario 2: Cafetería

Juan trabaja en unas oficinas del centro de Madrid. Todos los días, a las 11 de la mañana, tiene 30 minutos de descanso que utiliza para desayunar en una cafetería cercana.

Juan es usuario de la aplicación mWallet, que le permite utilizar varias modalidades de pago.

Cash Desk mediante NFC

Juan llega a la cafetería y realiza su pedido. Al finalizar, pide la cuenta al camarero y le indica que pagará usando mWallet y NFC.

Abre la aplicación, selecciona una tarjeta de su lista y acerca su teléfono a un dispositivo del establecimiento dotado de un lector NFC, realizándose el pago.

Juan podrá ver en su móvil un recibo con los datos de la operación.

Check In mediante geolocalización

A las 11 de la mañana, desde su mesa de trabajo, Juan inicia la aplicación mWallet desde la que puede ver aquellos establecimientos cercanos que permiten realizar Check In mediante geolocalización y entre los cuales se encuentra la cafetería en la que desayuna habitualmente.

Selecciona dicha cafetería, escoge la tarjeta con la que va a realizar el pago y confirma el Check In.

Se dirige al establecimiento, pide su desayuno y en el momento de pagar le basta con indicar al camarero su nombre para completar el pago. Recibirá en su dispositivo móvil el recibo de la operación.

Check In mediante QR

La cafetería en la que desayuna Juan tiene visible a los clientes un código QR que permite hacer Check In.

Cuando Juan llega al establecimiento, saca su teléfono móvil, abre la aplicación mWallet, selecciona una tarjeta de su lista y hace una foto del código QR utilizando el lector incorporado en la aplicación. Con esto se habrá realizado el Check In.

Tras desayunar, indica al camarero que ha hecho Check In, y basta con que se identifique para que quede realizado el pago.

En su dispositivo móvil aparecerá el recibo de la aplicación.

La tabla 10 indica las principales ventajas e inconvenientes de las modalidades de pago reflejadas en este escenario.

Cash Desk	Check In
<ul style="list-style-type: none"> + Su similitud con el pago tradicional con tarjeta de crédito hace que el usuario se sienta más cómodo y seguro. - El usuario debe guardar una fila hasta que llegue su turno para pagar. 	<ul style="list-style-type: none"> + El cliente, una vez terminada su consumición, solo debe indicar su nombre sin necesidad de sacar el móvil o la cartera para pagar. + Más rápido en el momento del pago. El usuario no tiene que sacar el dispositivo móvil. + Se puede hacer Check In desde el lugar en el que más cómodo y seguro se encuentre el usuario. + La eventual sensación de inseguridad que podría ocasionar el pago mediante Check In se reduce considerablemente al tratarse de establecimientos conocidos por el usuario. - Es necesaria una iluminación adecuada para que la cámara lea correctamente el código QR.

Tabla 10. Comparativa entre las modalidades de pago del Escenario 2

3.2.3 Escenario 3: Mercadillo

Cash Desk mediante QR

A David le gusta comprar en un mercadillo de antigüedades que se celebra los domingos en su ciudad.

Es habitual que no existan precios fijados y a los clientes les guste “regatear” con los vendedores.

David ha encontrado un reloj antiguo y tras acordar el precio con el vendedor, ambos deciden realizar la operación utilizando la aplicación mWallet.

El vendedor, utilizando su dispositivo móvil y la aplicación mWallet, genera un código QR con el precio del reloj y la cuenta en la que quiere que le sea abonado el importe.

A su vez, David, desde su Smartphone y la aplicación mWallet, selecciona una tarjeta de su lista y abre el lector de códigos QR incorporado en la misma. Captura el código que acaba de generar el vendedor. En este momento se muestra en la aplicación una pantalla en la que figuran los datos de la operación. Si son correctos y está de acuerdo, confirma la compra y se realiza el pago.

Este escenario no es el más idóneo para utilizar la modalidad de Check In.

La tabla 11 muestra las principales ventajas e inconvenientes de las modalidades de pago para el escenario 3.

Cash Desk	Check In
<ul style="list-style-type: none"> + Aprovecha la funcionalidad de “Comparte los gastos” para acordar un precio entre comprador y vendedor + Mayor sensación de seguridad. - Es imprescindible que el vendedor disponga de un móvil con la aplicación mWallet 	<ul style="list-style-type: none"> - Dificultad para establecer un servidor que almacene los datos de los check in - El mercadillo, por su carácter itinerante, no dispone de puntos concretos para realizar Check In - No ofrece seguridad en este escenario.

Tabla 11. Comparativa entre las modalidades de pago del Escenario 3

3.3. Análisis de requisitos

A partir de los escenarios descritos anteriormente, es posible extraer los requisitos que deberá tener la aplicación mWallet.

Tradicionalmente, dentro del área de la ingeniería del software, se identifican dos tipos de requisitos: los funcionales, que indican qué debe hacer el sistema, y los no funcionales, que indican las restricciones que tendrá el mismo.

En el diseño de la interacción es necesario comprender también la funcionalidad y las restricciones que tendrá el producto.

Sin embargo, en lugar de clasificar todos los requisitos en funcionales y no funcionales, se utilizarán las categorías consideradas en [18] y que son definidas a continuación.

- Requisitos funcionales
Especifican qué debe hacer el producto.
- Requisitos de datos
Indican el tipo, tamaño, persistencia, precisión y valor de los datos que utiliza la aplicación. Por ejemplo: en banca, los datos deben ser precisos y deben ser persistentes durante meses e incluso años. Además es información muy valiosa y debe ser tratada de forma segura.
- Requisitos de entorno
Especifican las circunstancias en las cuales se espera que el producto opere. Por ejemplo: la cantidad de luz, el ruido, el número de personas que accederán al sistema.
- Requisitos de perfil de usuario
Describen las características de los usuarios potenciales que utilizarán el producto.
- Requisitos de usabilidad
Especifican los objetivos de usabilidad y las medidas asociadas que requiere el producto, como por ejemplo: efectividad, eficiencia, seguridad, utilidad.

3.3.1 Requisitos funcionales

Acceso y registro

RFS-001	Registrar usuario
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Crear un nuevo usuario registrado a través de los formularios y el proceso de registro.
Justificación	El usuario debe tener una cuenta para poder realizar el acceso a la aplicación posteriormente. Esta cuenta de usuario proporciona mayor seguridad en el acceso a la aplicación

Tabla 12. Requisito funcional 001

RFS-002	Borrar usuario
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Eliminar los datos de un usuario registrado en el sistema
Justificación	El usuario deberá tener la opción de borrar su cuenta en cualquier momento.

Tabla 13. Requisito funcional 002

RFS-003	Iniciar sesión a través de PIN
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El usuario accederá a la aplicación a través de un PIN de 4 dígitos personal
Justificación	El usuario debe identificarse antes de poder acceder a la aplicación. Proporciona seguridad en caso de que otras personas puedan utilizar el dispositivo móvil

Tabla 14. Requisito funcional 003

RFS-004	Iniciar sesión a través de gesto/patrón
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El usuario accederá a la aplicación realizando un gesto sobre la pantalla
Justificación	Proporcionar una alternativa al acceso a través de PIN

Tabla 15. Requisito funcional 004

RFS-005	Cerrar sesión
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El usuario podrá cerrar la sesión
Justificación	Para garantizar la confidencialidad el sistema debe permitir cerrar la sesión de modo que se vuelva a necesitar el proceso de login para acceder

Tabla 16. Requisito funcional 005

Información del usuario

RFS-006	Modificar datos personales
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Se podrán añadir, modificar o eliminar datos personales del usuario
Justificación	El usuario puede cambiar de domicilio o haber cometido algún error en el proceso de registro. Debe ser capaz de mantener su información personal actualizada

Tabla 17. Requisito funcional 006

RFS-007	Crear PIN de acceso
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El usuario podrá crear o modificar el PIN de acceso a la aplicación
Justificación	El usuario tendrá configurado de manera personal un PIN que debe poder modificar en cualquier momento

Tabla 18. Requisito funcional 007

RFS-008	Añadir tarjeta de crédito
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Añadir datos de las tarjetas de crédito a través de formularios
Justificación	El usuario podrá añadir tantas tarjetas de crédito como desee para poder realizar sus pagos a través de la aplicación

Tabla 19. Requisito funcional 008

RFS-009	Añadir tarjeta de crédito a través de la cámara
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input checked="" type="checkbox"/> Opcional
Descripción	Añadir datos de las tarjetas de crédito mediante reconocimiento a través de la cámara del dispositivo móvil
Justificación	Hace más cómodo y sencillo el proceso de añadir una tarjeta a la aplicación

Tabla 20. Requisito funcional 009

RFS-010	Eliminar tarjeta de crédito
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Eliminar del sistema todos los datos asociados a una tarjeta introducida en la aplicación
Justificación	El usuario debe poder eliminar tarjetas que ya no utilice o no desee tener en la aplicación

Tabla 21. Requisito funcional 010

RFS-011	Añadir foto de perfil
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Incluye una foto al perfil del usuario
Justificación	Al realizar el Check In, junto con los datos del usuario se incluye su foto de perfil para que el establecimiento pueda verificar la identidad del cliente. Además proporciona al usuario una mayor personalización de la aplicación

Tabla 22. Requisito funcional 011

RFS-012	Cambiar la imagen de perfil
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Cambiar la foto de perfil establecida
Justificación	El usuario debe poder cambiar su foto de perfil en cualquier momento

Tabla 23. Requisito funcional 012

Métodos de pago e identificación del cliente

RFS-013	Pago Cash Desk mediante NFC
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El usuario realizará el pago acercando su dispositivo móvil a un lector NFC
Justificación	El usuario podrá usar NFC para poder realizar los pagos. Existen establecimientos que ya incorporan este método de pago.

Tabla 24. Requisito funcional 013

RFS-014	Pago Cash Desk mediante QR
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El usuario realizará el pago realizando una fotografía a un código QR en el momento del pago
Justificación	No todos los dispositivos cuentan con NFC sirviendo el código QR como alternativa de pago

Tabla 25. Requisito funcional 014

RFS-015	Pago mediante Check In
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El usuario enviará sus credenciales al entrar en el establecimiento y a la hora de realizar el pago únicamente se identificará.
Justificación	Es una alternativa al método de pago cash desk que, en algunos escenarios, puede ser de mayor utilidad

Tabla 26. Requisito funcional 015

RFS-016	Identificación a través de NFC en Check In
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Al acceder al establecimiento el usuario enviará sus credenciales a través de NFC
Justificación	El usuario debe identificarse al entrar en la tienda para poder realizar el pago mediante Check In

Tabla 27. Requisito funcional 016

RFS-017	Identificación a través de geolocalización en Check In
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El usuario podrá seleccionar una tienda cercana a su posición e identificarse en la misma sin necesidad de encontrarse físicamente en ella
Justificación	Mediante la geolocalización el usuario podrá conocer que establecimientos cercanos ofrecen el pago mediante Check In y realizar el mismo antes de llegar al establecimiento

Tabla 28. Requisito funcional 017

RFS-018	Identificación a través de código QR en Check In
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El usuario capturará un código QR situado a la entrada del establecimiento que le permitirá enviar sus credenciales
Justificación	Ofrecer alternativas al modelo de Check In en caso de que el dispositivo no cuente con NFC o geolocalización

Tabla 29. Requisito funcional 018

RFS-019	Anular Check In
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El usuario podrá cancelar un Check In activo borrando así toda su información del establecimiento
Justificación	Si el usuario no va a realizar ninguna compra debe poder cancelar la operación de Check In para asegurarse de que sus datos no son guardados en el establecimiento

Tabla 30. Requisito funcional 019

Cupones descuento y ofertas

RFS-020	Gestionar ofertas y descuentos
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	La aplicación mostrará ofertas y descuentos de los establecimientos
Justificación	El usuario puede ver que descuentos tiene de forma rápida y gestionarlos para realizar sus compras

Tabla 31. Requisito funcional 020

RFS-021	Aplicar ofertas y descuentos al pago
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Las ofertas y descuentos que desee utilizar el usuario serán aplicadas en el momento del pago
Justificación	El usuario puede gestionar en el momento del pago que ofertas de las disponibles quiere aplicar

Tabla 32. Requisito funcional 021

Historial de compras

RFS-022	Monitorizar transacciones
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Se mostrará un listado con las transacciones realizadas recientemente
Justificación	El usuario debe poder controlar las compras que ha realizado y gestionar así fácilmente sus gastos

Tabla 33. Requisito funcional 022

RFS-023	Monitorizar transacciones por tarjeta
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Se mostrará un listado con las transacciones realizadas recientemente con una tarjeta específica
Justificación	El usuario debe poder controlar las compras que ha realizado con una tarjeta concreta y gestionar así fácilmente sus gastos

Tabla 34. Requisito funcional 023

Transferencias

RFS-024	Generar códigos QR
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	La aplicación contará con un generador de códigos QR que permitirá a los usuarios crearlos para realizar transferencias
Justificación	Los usuarios podrán crear fácilmente códigos QR en los que se contendrá el identificador de la cuenta y el importe de la transferencia.

Tabla 35. Requisito funcional 024

RFS-025	Realizar transferencia (cash desk QR)
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	La aplicación permitirá que se realicen transferencias entre usuarios a través de un código QR
Justificación	Se deben poder realizar transferencias entre los usuarios usando las mismas tecnologías que para los pagos. De este modo se aprovecha la experiencia de los usuarios y se reduce la curva de aprendizaje

Tabla 36. Requisito funcional 025

3.3.2 Requisitos del entorno

REN-001	Adaptación de los establecimientos
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El establecimiento debe estar adaptado a algunos de los modelos de pago: Cash Desk o Check In (NFC, QR y Geolocalización)
Justificación	Los establecimientos deben contar con la tecnología usada por la aplicación mWallet

Tabla 37. Requisito de entorno 001

REN-002	Tecnología NFC
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El dispositivo móvil deberá disponer de NFC
Justificación	El usuario, para poder usar el pago mediante NFC, debe disponer de un dispositivo que posea esta tecnología

Tabla 38. Requisito de entorno 002

REN-003	Conexión a internet
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Se debe disponer de conexión a internet para poder utilizar la aplicación
Justificación	Los datos del usuario se encuentran en una base de datos. Es necesario conectarse a través de internet para poder acceder a ellos

Tabla 39. Requisito de entorno 003

REN-004	Acceso a la localización
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El dispositivo debe ser capaz de obtener la ubicación del usuario
Justificación	El móvil deberá tener activada la localización a través de GPS para que la aplicación pueda mostrar establecimientos cercanos que permitan el pago a través de mWallet

Tabla 40. Requisito de entorno 004

REN-005	Cámara de fotos
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El dispositivo debe de tener una cámara para poder leer los códigos QR
Justificación	Tanto las modalidades de Check In como cash desk cuentan con la posibilidad de realizar la interacción a través de la lectura de un código QR

Tabla 41. Requisito de entorno 005

3.3.3 Requisitos de datos

RDT-001	Base de datos
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	La información de los usuarios será almacenada en una base de datos
Justificación	La información de los usuarios (credenciales, tarjetas de crédito) debe ser accesible tanto por la aplicación como por el sistema instalado en los establecimientos. Esta información se encontrará almacenada en una base de datos.

Tabla 42. Requisito de datos 001

RDT-002	Cifrado de las cuentas bancarias
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Los datos bancarios de los usuarios se almacenarán de forma segura siguiendo un cifrado
Justificación	Los datos bancarios son información sensible por lo que debe garantizarse la confidencialidad de los mismos

Tabla 43. Requisito de datos 002

RDT-003	Cifrado de los datos personales
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Los datos personales de los usuarios se almacenarán de forma segura siguiendo un cifrado
Justificación	Los datos personales contienen información sensible y serán almacenados en la base de datos conforme a la Ley Orgánica de Protección de Datos de Carácter Personal.

Tabla 44. Requisito de datos 003

RDT-004	Almacenamiento de PIN de acceso
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El PIN de acceso personal a la aplicación será guardado en archivos internos de la aplicación que no pueden ser accedidos desde fuera de la misma
Justificación	El PIN de cada usuario se almacena en la aplicación ya que el acceso es más rápido que si éste fuese guardado en base de datos

Tabla 45. Requisito de datos 004

3.3.4 Perfil de usuario

RPU-001	Usuarios potenciales
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Los usuarios potenciales tendrán edades comprendidas entre los 18 y los 65 años
Justificación	Personas con acceso al mundo laboral y con cierta independencia económica y habituadas al uso de smartphones

Tabla 46. Requisito de perfil de usuario 001

RPU-002	Idioma
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	La aplicación se encontrará disponible en distintos idiomas para adaptarse a la lengua de cada usuario
Justificación	La aplicación podrá ser usada en cualquier establecimiento que soporte el pago mediante el sistema, independientemente del país

Tabla 47. Requisito de perfil de usuario 002

RPU-003	Familiaridad con el uso
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El usuario estará habituado al uso de un smartphone
Justificación	La aplicación será usada por personas que utilizan el móvil de forma habitual para realizar sus tareas

Tabla 48. Requisito de perfil de usuario 003

RPU-004	Uso habitual de tarjeta de crédito
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El usuario realiza habitualmente sus pagos mediante tarjeta de crédito
Justificación	Los usuarios de esta aplicación realizan sus pagos con tarjeta de crédito de forma habitual y están acostumbrados a ello.

Tabla 49. Requisito de perfil de usuario 004

3.3.5 Requisitos de usabilidad

RUS-001	Sencillez de uso
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	La aplicación debe ser sencilla de usar
Justificación	El usuario debe centrarse en la tarea y no en el uso de la aplicación

Tabla 50. Requisito de usabilidad 001

RUS-002	Aprendizaje rápido
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	La aplicación requerirá muy poco aprendizaje para ser usada
Justificación	Debe ser intuitiva para que los usuarios no necesiten aprender a usar la aplicación. Para ello se usarán mensajes e iconos descriptivos

Tabla 51. Requisito de usabilidad 002

RUS-003	Rapidez en los pagos
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	El pago a través de la aplicación debe ser rápido
Justificación	El pago mediante la aplicación debe llevar el mismo o menor tiempo que si se realizase con tarjeta normal.

Tabla 52. Requisito de usabilidad 003

RUS-004	Prevención de errores
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	La aplicación debe evitar que se produzcan errores de los usuarios. Se usarán mensajes descriptivos y se proporcionarán acciones para cancelar o volver atrás
Justificación	La información con la que se trabaja es muy sensible. Hay que evitar que se realicen pagos indeseados.

Tabla 53. Requisito usabilidad 004

RUS-005	Interfaz efectiva
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Utilizar recursos de programación que faciliten la navegación, la cumplimentación de formularios y la realización de las tareas principales de la aplicación.
Justificación	La interfaz debe ser clara y sencilla para que el usuario pueda realizar las tareas de forma rápida y con poco esfuerzo

Tabla 54. Requisito usabilidad 005

RUS-006	Evitar los campos de texto libre
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Descripción	Utilizar desplegables y “radio buttons” siempre que sea posible para evitar errores de los usuarios
Justificación	Los campos de texto libre son una fuente común de errores en los usuarios.

Tabla 55. Requisito usabilidad 006

4. Diseño e implementación

4.1 Introducción y proceso de diseño

En este capítulo se describe la aplicación móvil desarrollada en este proyecto, mWallet, la cual ayuda a los usuarios a realizar los pagos en diferentes contextos.

El objetivo principal buscado es desarrollar una interfaz útil y efectiva. Para conseguirlo, se ha seguido un Diseño Centrado en el Usuario (figura 2), estudiando los usuarios potenciales y teniéndolos en consideración durante todo el proceso de diseño. Se han analizado distintos escenarios donde puede ser utilizada mWallet y a partir de ellos se han obtenido los requisitos que deberá cumplir la aplicación para proporcionar toda la funcionalidad.

Además, se han analizado las últimas tecnologías móviles así como interfaces de usuario avanzadas que tratan aspectos similares.

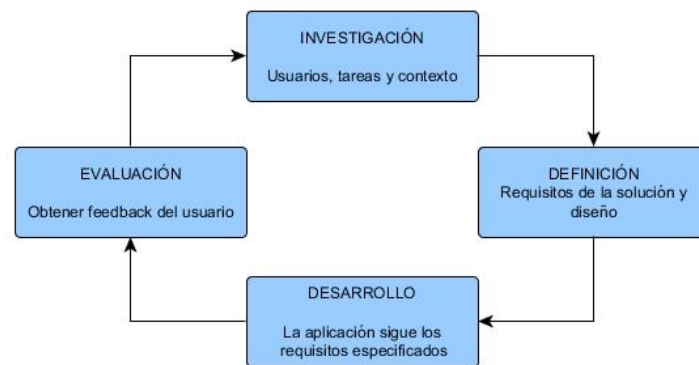


Figura 2. Proceso de diseño centrado en el usuario

En el apartado 4.2 *Tecnología*, se explican las tecnologías NFC y QR, que son utilizadas para implementar los modelos de interacción vistos en el capítulo 3.

El apartado 4.3 *Workflow: Diseño de la interacción*, describirá el flujo de trabajo de la aplicación para las distintas metodologías de pago.

En el apartado 4.4 *Diseño de la interfaz* se muestra el proceso seguido para realizar la interfaz de la aplicación móvil. Incluyendo wireframes y mockups de las distintas pantallas que la componen, y comparándolas con el diseño final.

Por último, en el apartado 4.5 *Detalles de implementación*, se explicará la arquitectura del sistema, el diseño de la base de datos y se detallarán las librerías utilizadas para el desarrollo del sistema, tanto de la aplicación móvil como del lado servidor.

4.2 Tecnología

4.2.1 RFID

RFID (Radio Frequency Identification) [19] es una tecnología de comunicación inalámbrica que utiliza ondas de radio para transmitir la información desde una etiqueta RFID a un lector que la procesa.

La arquitectura de un sistema RFID consta de dos componentes principales:

- Etiqueta RFID
Las etiquetas RFID actúan como transpondedores y contienen la información que debe leerse.
Están formadas por tres partes: una antena, un transductor de radio y un microchip. La antena se encarga de transmitir al lector la información contenida en el microchip.
- Lector RFID:
El lector RFID actúa como transceptor (transmisor y receptor). Éste emite señales de forma periódica para tratar de detectar las etiquetas cercanas.
Cuando una etiqueta es detectada, el lector recibe la información emitida por la misma y la envía al sistema que se encargará de procesarla. Este sistema puede ser, por ejemplo, una base de datos.

En la figura 3 se puede ver la estructura de un sistema RFID.

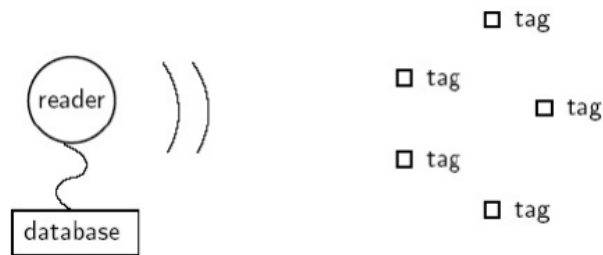


Figura 3 Arquitectura de un sistema RFID

Las etiquetas RFID a su vez pueden clasificarse en tres tipos según la fuente de energía que utilicen [20]:

- Etiquetas pasivas
Este tipo de etiquetas no tienen una fuente de alimentación interna, la energía es suministrada por el lector.

Cuando la antena que contiene la etiqueta recibe la señal del lector se genera un campo electromagnético que produce la energía necesaria para que la información contenida pueda ser transmitida.

La tabla 56 muestra las ventajas y desventajas de este tipo de etiquetas.

Ventajas	Desventajas
<ul style="list-style-type: none"> ▪ Coste bajo ▪ Tiempo de vida alto debido a que no tienen una batería ▪ Son pequeñas por lo que pueden ser utilizadas en gran cantidad de objetos ▪ Mayor sensibilidad 	<ul style="list-style-type: none"> ▪ Rango de actuación corto. Necesitan que el lector esté a unos pocos metros ▪ Menores tasas de transmisión ▪ Capacidad de almacenamiento reducida (128 bytes)

Tabla 56. Ventajas y desventajas de las etiquetas pasivas

▪ Etiquetas activas

Las etiquetas activas, a diferencia de las pasivas, se encuentran equipadas con una batería interna que da energía al microchip y permite enviar señales al lector.

La mayor ventaja que tiene este tipo de etiquetas frente a las pasivas es que el rango de actuación es mucho mayor, pudiendo llegar a decenas o cientos de metros.

La tabla 57 muestra las ventajas y desventajas de las etiquetas activas.

Ventajas	Desventajas
<ul style="list-style-type: none"> ▪ Mayor rango de actuación. ▪ Puede tener sensores integrados ▪ Altas tasas de transmisión ▪ Mayor capacidad de almacenamiento ▪ Puede iniciar la comunicación 	<ul style="list-style-type: none"> ▪ Mayor coste que las pasivas ▪ Tamaño grande ▪ La vida útil de la etiqueta queda limitada a la de la batería que contiene ▪ Tiene un coste de mantenimiento si se pueden reemplazar las baterías

Tabla 57. Ventajas y desventajas de las etiquetas activas

▪ Etiquetas semi-pasivas

Combinan características de los dos tipos anteriores. Tienen una batería interna que se encarga de alimentar el chip y además utilizan la energía del lector para realizar la transmisión de los datos.

Un tipo especial de la tecnología RFID es el NFC (Near Field Communication) [21]. El NFC fue desarrollado conjuntamente por Philips y Sony a finales del año 2002 para realizar comunicaciones “contactless” entre dispositivos.

Como es un tipo de RFID, los dispositivos que participan en la comunicación pueden ser activos o pasivos.

Existen 3 tipos de dispositivos NFC que pueden formar parte de la comunicación:

- Dispositivos móviles con NFC (móvil NFC)
- Etiquetas NFC
- Lectores NFC

Estos dispositivos pueden tomar dos roles en la comunicación: el *iniciador* se encarga de mandar peticiones al *receptor* (target) y este último se encargará de responder a las solicitudes recibidas.

Según el rol que tome cada dispositivo pueden existir tres modos de operación que se resumen en la tabla 58.

Modo de operación	Iniciador	Receptor
Lector/Escritor	Móvil NFC	Etiqueta NFC
Peer to peer	Móvil NFC	Móvil NFC
Emulación de tarjeta	Lector NFC	Móvil NFC

Tabla 58. Modos de operación NFC

En los tres casos, la interacción es similar: se acerca el dispositivo que actúa como iniciador al receptor; cuando se encuentran próximos el iniciador solicitará al receptor la información deseada o en el caso del modo de operación Lector/Escritor puede ser una solicitud de escritura sobre la etiqueta.

La tecnología NFC es utilizada en multitud de áreas y para realizar una gran variedad de tareas. Algunos de los usos que tiene el NFC son los siguientes:

- Aplicaciones de salud: cuidado de personas dependientes, monitorización de los pacientes.
- Espacios inteligentes: utilizan etiquetas NFC repartidas por el entorno (domótica).
- Aplicaciones para intercambiar y compartir información.
- Sistemas de pago a través del Smartphone.
- Aplicaciones para el entretenimiento.
- Redes sociales
- Aplicaciones educativas

4.2.2 Códigos QR

Un código QR (Quick Response, respuesta rápida) [22] es un código de barras bidimensional donde la información se encuentra codificada, tanto de forma vertical como horizontal (figura 4). Esto hace que puedan almacenar una mayor cantidad de datos (hasta 7089 caracteres, siendo estos dígitos) que los códigos de barras tradicionales (máximo 20 dígitos).



Figura 4 Código QR. [22]

Los datos que pueden almacenar estos códigos son:

- Números (0 – 9)
- Alfanuméricos
- Caracteres Kanji
- Datos binarios

Otra ventaja de los códigos QR sobre los códigos de barra tradicionales es que los primeros pueden ser leídos desde cualquier posición. Un lector de QR es capaz de determinar la orientación correcta del código gracias a los 3 cuadrados que se encuentran situados en las esquinas.

Los códigos QR están divididos en varias partes (figura 5), cada una de las cuales tiene una función.

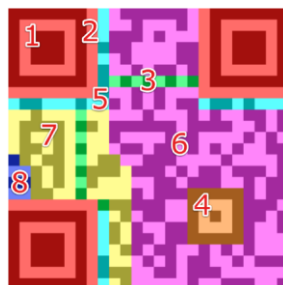


Figura 5 Partes del Código QR [22]

Los **patrones de posición (1)** se encuentran en tres esquinas del código y permiten al lector determinar la posición correcta del código.

Alrededor de los patrones de posición se encuentran los **separadores (2)** que permiten distinguir entre los datos y el patrón.

Los **patrones de sincronización (3)** indican el ancho del código. Si el código es muy grande se pueden llegar a incluir **patrones de alineamiento (4)** para facilitar al lector la decodificación del QR.

El resto de partes contienen los datos e información para el control de errores.

Los códigos QR pueden contener una gran variedad de información, lo que permite que puedan tener muchos usos [23]:

- Información de contactos (Utilizados en tarjetas de contactos para que simplemente con escanear el código el contacto sea incluido en el teléfono).
- Eventos de calendario
- Direcciones de correo electrónico
- Número de teléfono móvil
- Geolocalización (Automáticamente al escanear el código se abre la aplicación de mapas del dispositivo con la dirección incluida en el código QR).
- SMS
- Texto
- Configuración de una red WiFi
- URL (abren automáticamente el navegador iniciando la página contenida en el código).

4.3 Workflow: Diseño de la interacción

En este apartado se van a detallar los pasos que sigue el usuario para realizar cada una de las modalidades de pago presentadas en el capítulo 3.

Para cada modalidad se compararán las diferencias entre las distintas tecnologías que pueden ser utilizadas: NFC, QR y geolocalización.

4.3.1 Cash Desk

Dependiendo de la tecnología que se utilice, QR o NFC, la interacción tendrá algunas variaciones que se explican a continuación.

NFC

En la modalidad de cash desk con NFC el usuario, tras seleccionar una tarjeta válida de su lista de tarjetas de la aplicación, acerca el dispositivo móvil a un lector NFC situado en el establecimiento.

Este lector valida la tarjeta y se procede a realizar el pago. En el caso de que el usuario tenga alguna oferta disponible a aplicar en su compra se le mostrará una pantalla previa al pago donde puede seleccionar si desea utilizarla o no.

Una vez realizado el pago, el usuario podrá ver en su smartphone el recibo con la información detallada de la transacción que acaba de realizar. En este caso el recibo se muestra con una animación que simula la impresión de un ticket en el mundo real. La figura 6 muestra este flujo de trabajo.

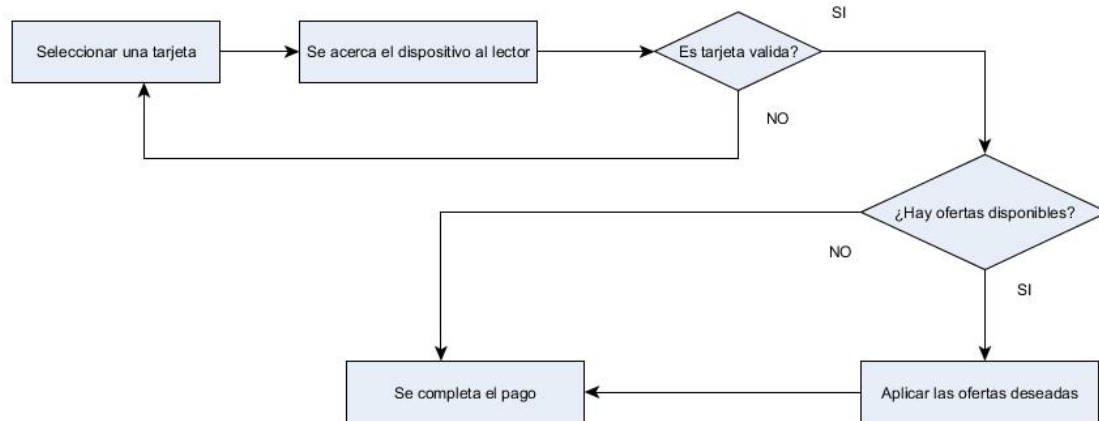


Figura 6. Flujo de trabajo de la modalidad de pago Cash Desk usando NFC

QR

Al realizar la compra, en el recibo que se entrega al usuario se incluye un código QR.

El usuario, al igual que con el NFC, selecciona una tarjeta de la lista. A continuación, abre la opción de lectura de QR y hace una captura del código del recibo.

Si existe algún descuento aplicable a la compra se muestra una pantalla al usuario donde podrá seleccionar si desea utilizarla o no.

Finalmente se procesa el pago y se muestra un recibo en la pantalla del dispositivo móvil con toda la información de la transacción que se acaba de realizar. La figura 7 muestra este flujo de trabajo.

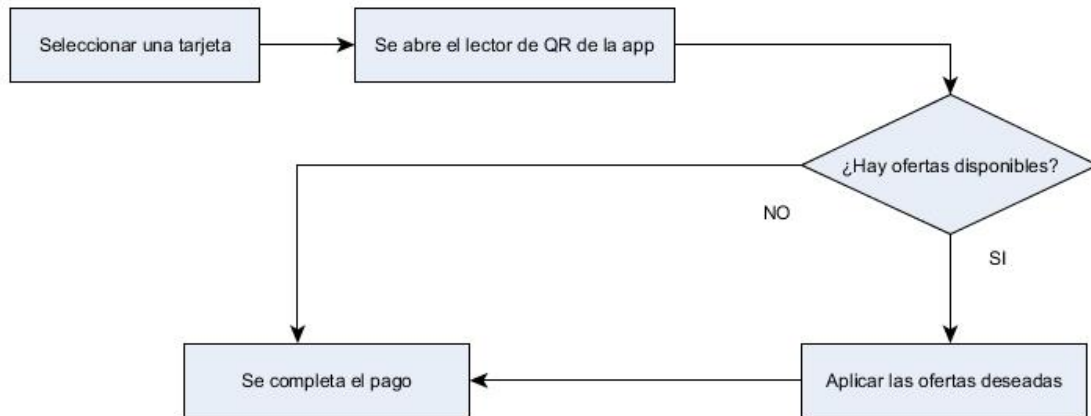


Figura 7. Flujo de trabajo de la modalidad de pago Cash Desk usando QR

4.3.2 Check In

Esta modalidad podrá ser realizada de tres formas distintas, a través de QR, geolocalización o NFC.

QR

El usuario selecciona la tarjeta con la que desea realizar la compra. Una vez escogida abre el lector de QR incluido en la aplicación y fotografía el código que se encontrará en el establecimiento (puede estar por ejemplo: en la carta, en el escaparate etc.).

A continuación se muestra en la pantalla del dispositivo un resumen con los datos del establecimiento y los datos personales del usuario. En el caso de que existan ofertas disponibles también se indican. La figura 8 muestra gráficamente los pasos de este flujo de trabajo.

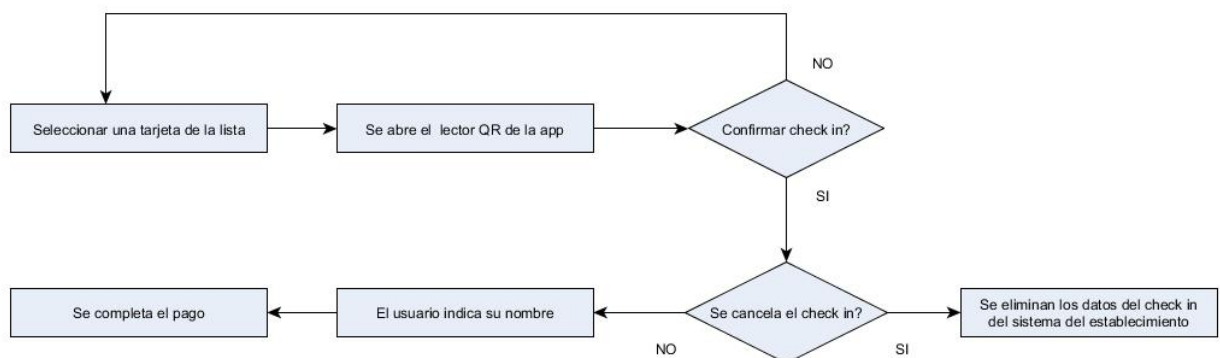


Figura 8. Flujo de trabajo de la modalidad de pago Check In usando QR

Geolocalización

Desde la sección “Explora” el usuario puede ver que establecimientos cercanos admiten el pago a través de mWallet.

Al pulsar sobre un establecimiento se mostrarán los datos de este y se permitirá al usuario seleccionar una tarjeta y realizar el check in. La figura 9 muestra este flujo de trabajo.

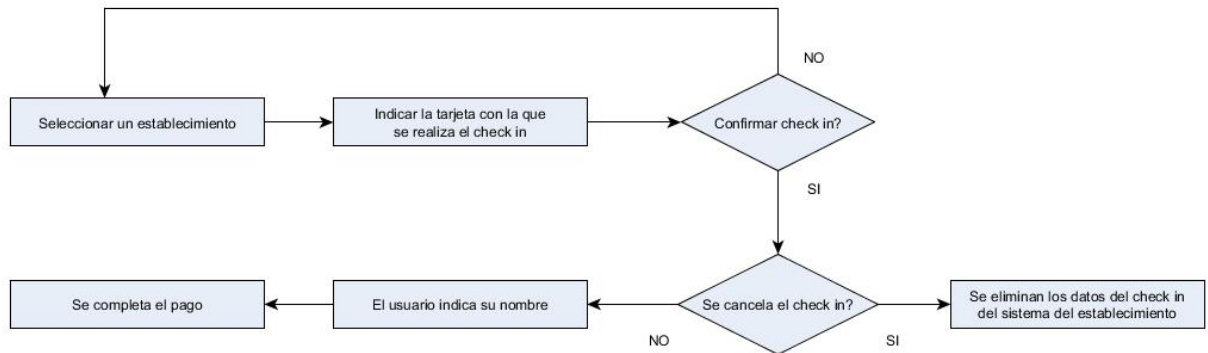


Figura 9. Flujo de trabajo de la modalidad de pago Check In usando geolocalización

NFC

La tercera forma de realizar check in es a través del NFC. La interacción es similar a la que se realizaba en el cash desk. El usuario selecciona la tarjeta y acerca el dispositivo al lector NFC. Cuando se lee la información, el lector envía al establecimiento los datos del check in del usuario.

En este caso, otra posible interacción sería que a la entrada del establecimiento se instalen lectores NFC y automáticamente, siempre y cuando el usuario habilite esta opción, al acceder un cliente con la aplicación, se realice el check in utilizando la tarjeta por defecto. La figura 10 muestra los pasos seguidos.

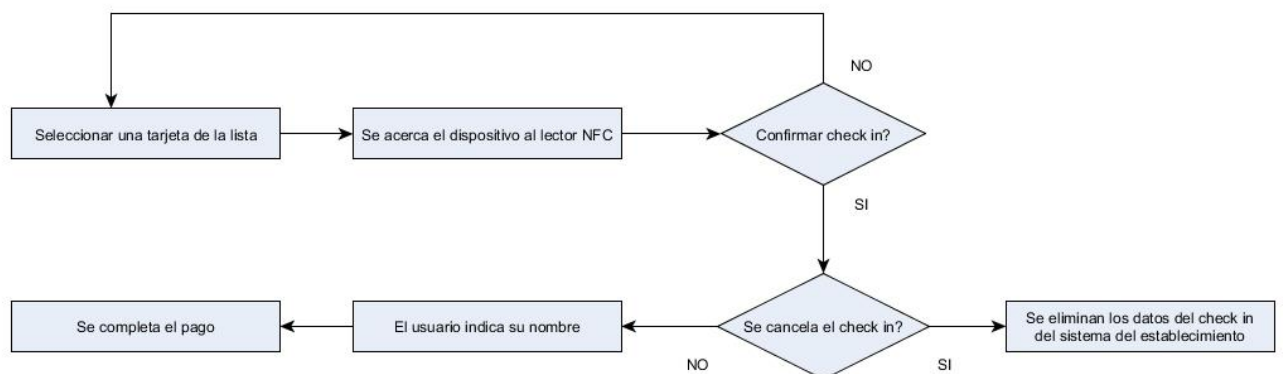


Figura 10. Flujo de trabajo de la modalidad de pago Check In usando NFC

En los tres casos el procedimiento para realizar el pago es el mismo. El usuario validará el check in, pudiendo ser éste cancelado en cualquier momento.

Al realizar la compra el cliente sólo deberá indicar al personal del establecimiento el nombre con el que realizó el check in y se procederá a completar el pago.

Una vez realizado el check in, en la “Cartera” se muestra una barra informando al usuario de que tiene activo un check in. Mientras esté pendiente no se permite realizar ninguna otra transacción con la aplicación.

Si el usuario desea pagar con cash desk o realizar un nuevo check in debe cancelar el que se encuentra activo en ese momento.

Para cancelar el check in, el usuario puede pulsar sobre la barra que aparece en la “Cartera” o bien puede acceder a la sección “Actividad reciente”, desde donde podrá ver la información del mismo y cancelarlo.

Al completar el check in, dependiendo del estado de la aplicación, pueden darse dos interacciones diferentes:

- Si la aplicación está abierta y en primer plano
En este caso se muestra automáticamente un recibo al usuario al igual que ocurría en el caso de cash desk.
- Si la aplicación está cerrada o no se encuentra en primer plano
En este escenario el dispositivo muestra una notificación al usuario. Al pulsar sobre la misma se abre el recibo del check in realizado.

4.3.3 Transferencias: Comparte los gastos

mWallet permite que una persona complete una compra con una de sus tarjetas y posteriormente comparta ese gasto con otros usuarios de la aplicación.

Desde la sección “Comparte los gastos” se puede generar un código QR que puede ser leído por otros usuarios de mWallet de tal modo que estos realicen una transferencia a la cuenta especificada en el código.

A continuación se describe el flujo de trabajo para esta interacción, distinguiendo entre los dos actores que participan:

- Usuario al que se va a realizar la transferencia
 1. El usuario accede a la sección “Comparte los gastos” desde el menú lateral de la aplicación.
 2. Rellena el formulario indicando la tarjeta a la que debe hacerse la transferencia y el importe de la misma.
 3. Pulsará sobre el botón “Generar código” para crear el código QR.
 4. Mostrará el código a los usuarios que vayan a realizar la transferencia.

- Usuario que realiza la transferencia
 1. Se selecciona una tarjeta desde la “Cartera” y se pulsa sobre el icono del lector de códigos QR
 2. Se captura el código al igual que sucedía en la modalidad de pago con cash desk y QR.
 3. Se muestra una pantalla con la información de la transferencia: tarjeta desde la que se va a realizar, tarjeta de destino e importe.
 4. Se valida la transferencia y se muestra un recibo al usuario informándole de la transacción

4.4. Diseño de la interfaz móvil

Como parte del proceso de diseño de la aplicación mWallet se han realizado wireframes y mockups de las distintas pantallas que la componen.

A través de los wireframes de baja fidelidad se puede realizar un primer diseño de la aplicación de manera rápida y que servirá como base para los desarrollos posteriores. A partir de estos wireframes se ha creado el mockup de la aplicación.

El mockup proporciona una visión más parecida al resultado que finalmente se obtendrá. Por este motivo suele ser utilizado para obtener un feedback de los clientes así como otras personas interesadas en la aplicación, como por ejemplo, los desarrolladores.

En este apartado se describirán cada una de las pantallas que conforman el prototipo de la aplicación. Además, se realizará una comparativa entre el mockup y el diseño final como se puede ver en la *figura 11*.

Para elaborar las interfaces se han seguido las guías de diseño de la página oficial de desarrolladores de Android [24].

Android es uno de los sistemas operativos más extendidos en la actualidad y además dispone de herramientas y código abierto que hacen más fácil el desarrollo de prototipos como el elaborado. En el caso del sistema operativo iOS es necesario disponer de una licencia de pago para poder instalar las aplicaciones en el dispositivo móvil. Esto motivó la elección de Android para este proyecto.

En el anexo III - Interfaz de mWallet se muestran las capturas de pantalla de la aplicación final.

4.4.1 Registro

La primera vez que se accede a la aplicación se pide al usuario que introduzca sus datos personales, nombre y apellidos, y sus datos de acceso a mWallet, email y contraseña, que le servirán para poder acceder a su cuenta desde otro dispositivo distinto al que la instaló por primera vez.

En la primera pantalla del registro (*figura 11*) se muestra el formulario para introducir la información mencionada anteriormente. Siguiendo los patrones de seguridad en web y aplicaciones el campo de la contraseña del usuario se muestra mediante puntos, evitando así que la contraseña pueda verse en claro.

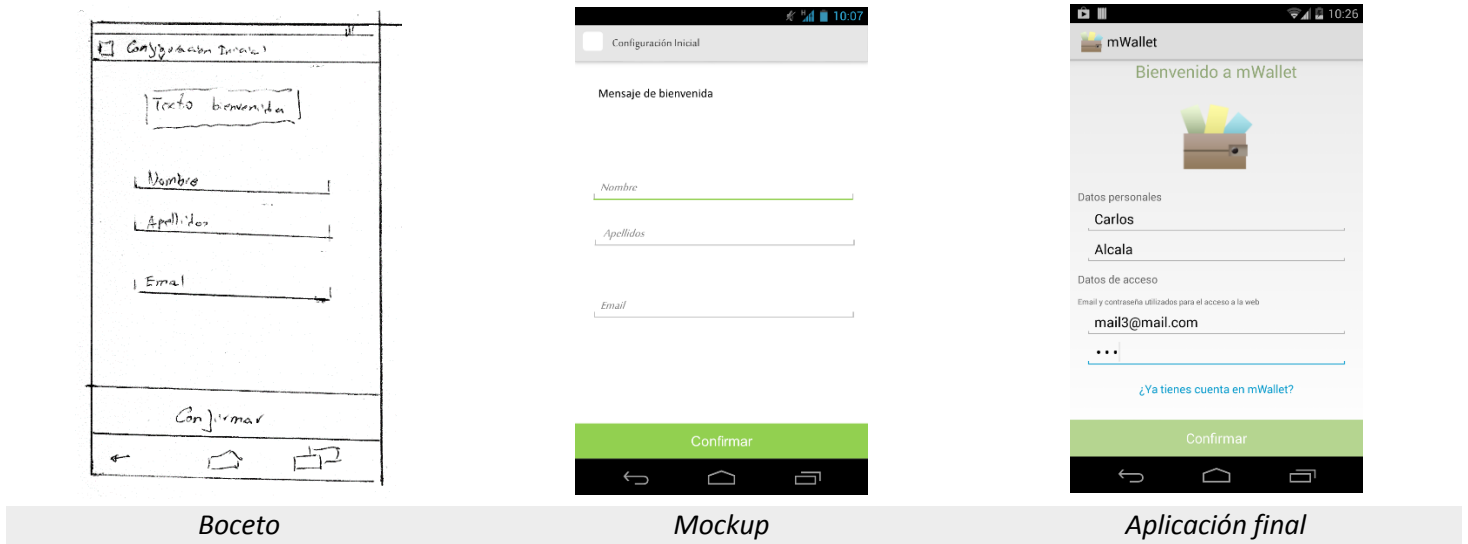


Figura 11. Proceso de diseño de la pantalla de registro

Tras completar este paso, aparece una nueva pantalla que permite al usuario establecer su imagen de perfil. Se activa por defecto la cámara frontal en caso de que el dispositivo cuente con una, evitando así que el usuario tenga que cambiar la cámara de forma manual.

La foto puede ser repetida tantas veces como el usuario desee.

En el último paso del registro se pide configurar el PIN de acceso. Este es el PIN que introducirá el usuario cada vez que abra la aplicación, proporcionando mayor seguridad a ésta.

4.4.2 Acceso a la aplicación

El acceso a la aplicación se hace a través de un PIN de seguridad de 4 dígitos que es configurado por el usuario al abrirla por primera vez. Este PIN también puede ser modificado en cualquier momento desde la pantalla de configuración.

Para el teclado numérico se ha empleado un diseño similar al utilizado en otras aplicaciones o en las pantallas de desbloqueo tanto de Android como de iOS. De este modo se aprovecha el user experience y el usuario no necesitará aprender su funcionamiento.

Según se van introduciendo los dígitos del PIN se van rellenando los círculos situados sobre el teclado en color verde, pero sin mostrar el PIN en claro por motivos de seguridad.

Si la contraseña introducida es incorrecta, se dará al usuario un feedback visual mediante una animación en los círculos, que indica que se ha producido un error.

La figura 12 muestra el proceso de elaboración de la pantalla de acceso a la aplicación.

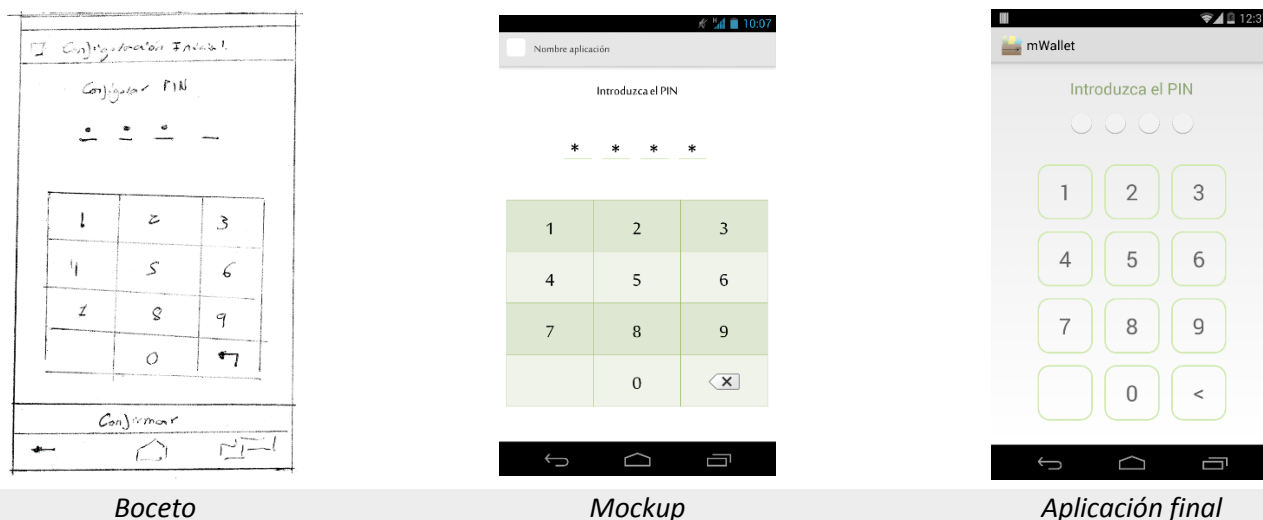


Figura 12. Proceso de diseño de la pantalla de acceso a la aplicación

4.4.3 Pantalla Principal. Cartera

Una vez realizado el acceso a la aplicación la primera sección que se encuentra es la “Cartera” (figura 13). Ésta contiene la lista de tarjetas añadidas por el usuario y con las que podrá realizar los pagos.

La cartera se muestra en primer lugar ya que el objetivo principal de la aplicación es simplificar y facilitar la tarea de realizar los pagos. Con solo abrir la aplicación el usuario ya podrá pagar, reduciéndose así el número de pasos necesarios para este fin.

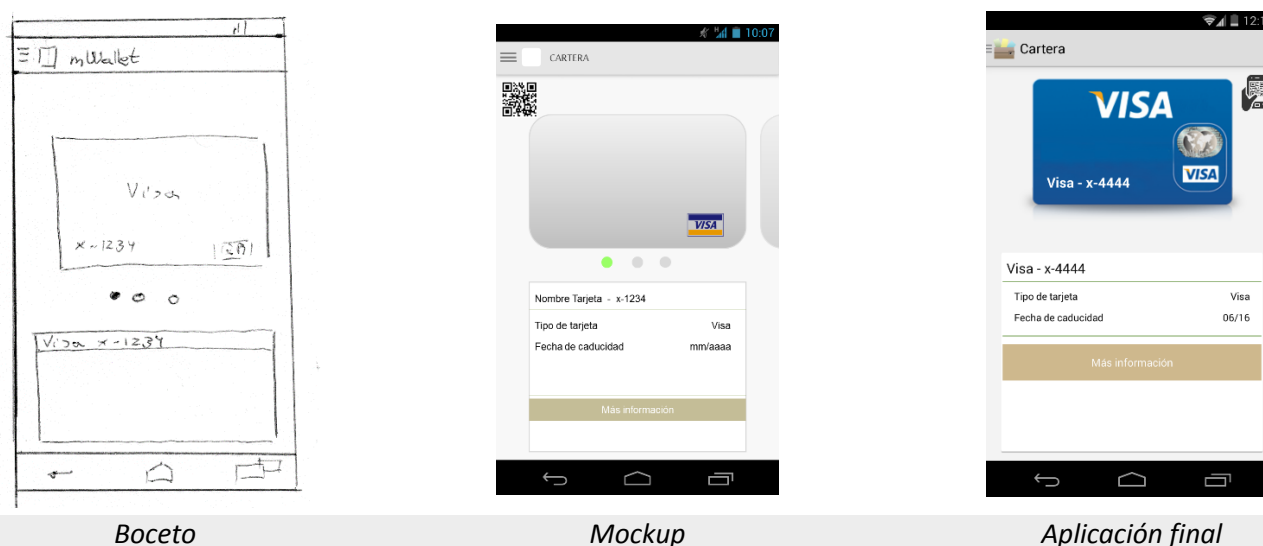


Figura 13. Proceso de diseño de la sección Cartera

Desde esta sección el usuario debe ser capaz de usar cualquiera de las modalidades de pago que ofrece la aplicación y que se explicaron en el capítulo 3.

Para poder leer un código QR se incluye un icono que indica al usuario que debe pulsar sobre él para abrir el lector QR.

En el boceto inicial se propuso incluir sobre la tarjeta un texto que indicaba que pulsando sobre la misma se activaría el lector de códigos QR. Sin embargo, esta disposición podía no ser interpretada de forma correcta. En el diseño final se decide añadir el icono a un lateral de la tarjeta, lo que resulta visualmente más efectivo y al mismo tiempo sigue quedando claro que la lectura del código QR está asociada a esa tarjeta.

Desde esta sección el usuario también podrá ver información básica de la tarjeta, de modo que pueda identificarla de forma sencilla. No obstante, no se muestra toda la información por motivos de seguridad.

Siguiendo el requisito 3.3 de seguridad de la normativa Payment Card Industry Data Security Standard [34] (PCI DSS), solo los 6 primeros dígitos de la tarjeta y los cuatro últimos pueden ser mostrados. En la aplicación desarrollada solo serán enviados al usuario los 4 últimos dígitos y se visualizará siguiendo el formato: Tipo de tarjeta - x- dddd; siendo dddd los últimos 4 dígitos.

El resto de información que puede ser mostrada en claro es el titular de la tarjeta y la fecha de caducidad de la misma.

Así mismo, existe un botón denominado “Más Información” que permite acceder a una nueva pantalla en la que se refleja de un modo más detallado los datos de la tarjeta y los últimos movimientos de la misma.

4.4.4 Navegación

En Android existen múltiples formas de diseñar la navegación entre las distintas secciones de la aplicación. Antes de obtener el diseño final, en el que se usa el patrón de diseño “Navigation Drawer”, se realizó una comparativa entre las distintas opciones:

- Pantalla inicial con iconos
 - La pantalla principal de la aplicación mostraría todas las secciones de la misma pudiéndose acceder a cada una de ellas pulsando sobre un icono.
 - Este diseño presenta desventajas con respecto al utilizado ya que para realizar los pagos el usuario necesitaría efectuar un paso extra para acceder a la cartera.
- Navegación a través de pestañas
 - Se muestran en la parte superior de la pantalla todas las secciones de la aplicación. El usuario podrá navegar tocando sobre las pestañas.
 - La navegación se realiza de forma rápida y no se añaden pasos extra como ocurría con el modelo anterior. Sin embargo, en dispositivos con pantallas pequeñas no es posible visualizar todas las pestañas, quedando algunas ocultas. El objetivo que se busca es que el usuario pueda saber en todo momento que tareas puede realizar.

- Las guías de diseño de Android [24] aconsejan el uso de pestañas si:
 - El número de secciones no es superior a 3. En el caso de mWallet se tienen un total de 7.
 - Se desea que el usuario sea consciente en todo momento de las distintas secciones disponibles. Al tener mWallet un total de 7, muchas de ellas quedarían ocultas y el usuario no podría ver todas al mismo tiempo.
- Navigation Drawer [36]
 - Es un panel que se encuentra oculto en el lateral izquierdo de la pantalla. El usuario puede acceder a él en todo momento bien deslizando el dedo desde el lateral izquierdo o pulsando sobre el icono de la aplicación que se encuentra situado en el action bar (accesible desde toda la aplicación).
 - Esta forma de navegación es aconsejada según las guías de diseño si se tiene un alto número de secciones y se quiere que el usuario pueda acceder de forma directa a cualquiera de las mismas independientemente de la pantalla en que se encuentre.

En el menú lateral se muestran las secciones de la aplicación junto con un icono representativo de cada una de ellas. Se sigue así la heurística de Nielsen [37] “reconocimiento antes que recuerdo”; de modo que el usuario, con solo ver el icono, identifica el contenido de cada sección sin necesidad de recordarlo.

En el diseño final de la aplicación se decide incluir en la parte superior del menú la foto y el nombre del usuario, consiguiendo así que ésta sea más cercana y personal.

La figura 14 muestra el proceso de diseño de la navegación a través de mWallet.

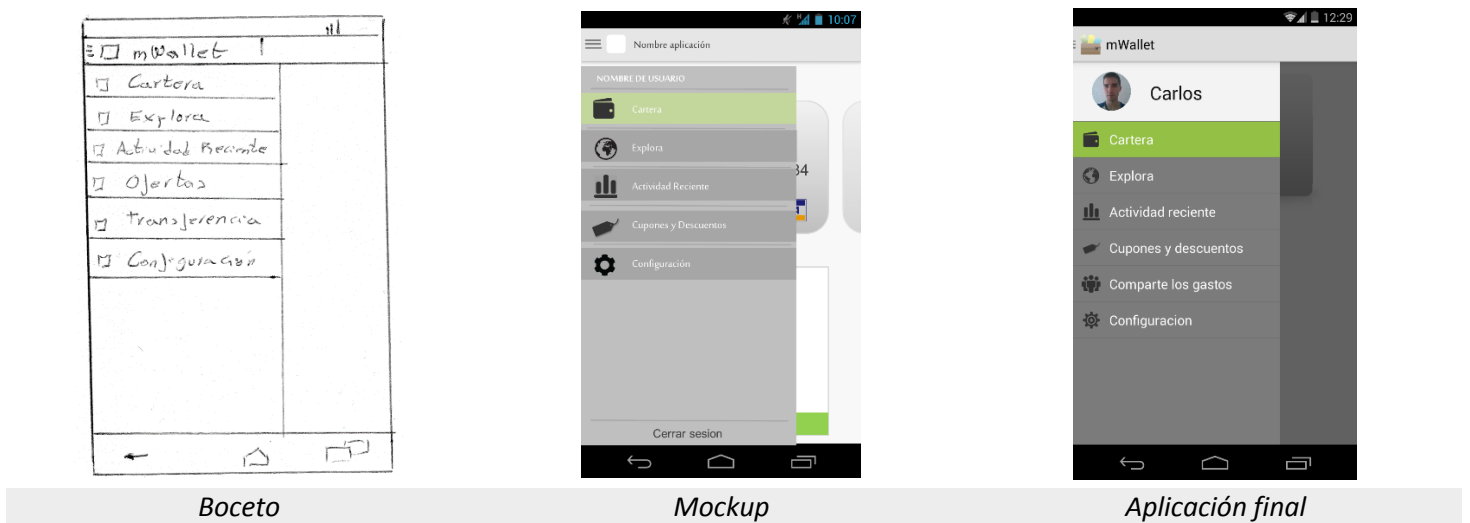


Figura 14. Proceso de diseño del menú lateral de navegación

4.4.5 Formulario de nueva tarjeta

El usuario podrá añadir una nueva tarjeta a través de un formulario (figura 15) en el que se le solicitará que introduzca los datos de la tarjeta y la información de facturación.

En los diseños iniciales se consideró que todos los campos fuesen de texto. Sin embargo, esto presentaba problemas, ya que campos como el número de la tarjeta o la fecha debían seguir un formato específico.

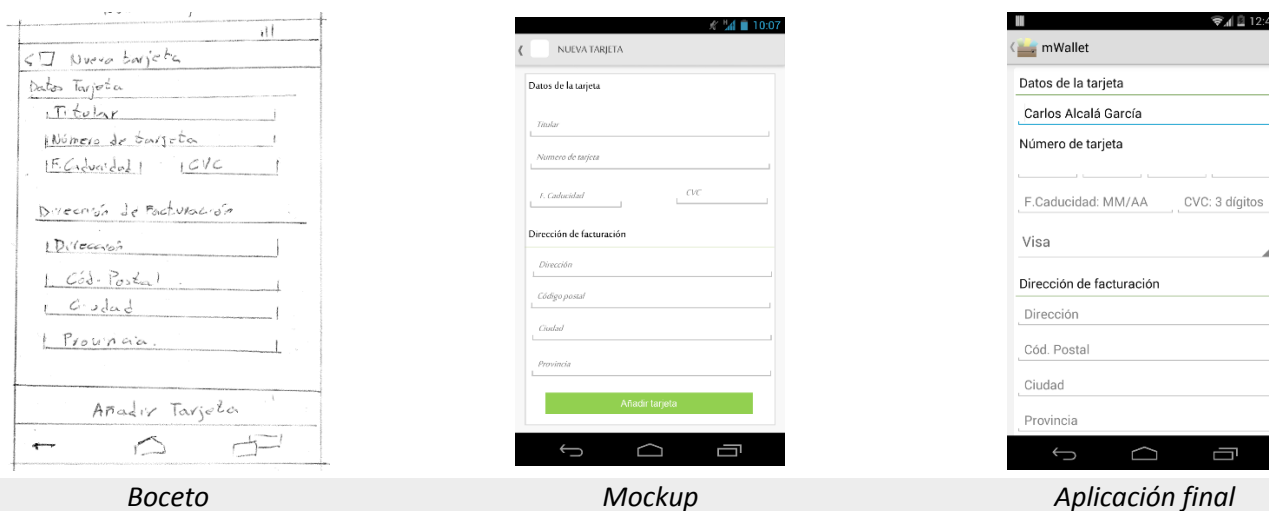


Figura 15. Proceso de diseño del formulario de nueva tarjeta

Para evitar que el usuario pueda cometer errores, en el diseño final se implementan los siguientes cambios:

- Se utiliza un campo desplegable para seleccionar el tipo de tarjeta (visa, mastercard). De este modo el usuario solo podrá indicar uno de los tipos válidos y soportados por la aplicación.
- Se divide el campo único del número de la tarjeta en cuatro campos distintos de cuatro dígitos cada uno. Cuando el usuario alcanza el final de un campo (introduce el cuarto dígito) automáticamente salta al siguiente. Se permite así que el usuario pueda introducir fácilmente el número, lo pueda ver de forma más clara y no tenga que estar pendiente del formato que debe seguir.
- La fecha de caducidad debe seguir el formato *mm/aa*. Si se deja como un campo de texto libre, el usuario puede introducir la fecha sin seguir la estructura válida para la aplicación. Para simplificar el control de errores y la tarea del usuario se ha decidido implementar en el diseño final el selector (figura 16) de mes y año que se muestra a continuación:



Figura 16. Selector de fecha de caducidad incluido en el diseño final

Mediante este selector se evitan los problemas comentados anteriormente y el usuario puede introducir de forma más rápida y sencilla la fecha de caducidad.

Para facilitar la cumplimentación del formulario al añadir sucesivas tarjetas se ha establecido que por defecto aparezca relleno el campo “Titular” con el nombre y apellidos del usuario registrado.

Otra consideración que se ha tenido en cuenta a la hora de diseñar este formulario es la variedad de tamaños de pantalla en los que la aplicación puede ser utilizada. Debido a que es un formulario largo, es posible que en pantallas muy pequeñas parte de la información no fuese visible.

Para que la aplicación pueda ser usada en cualquier dispositivo, independientemente del tamaño de su pantalla, se ha implementado el formulario con un scroll view, de modo que el usuario puede deslizar hacia arriba o hacia abajo la pantalla para ver el formulario completo.

Siempre que el usuario quiera añadir una nueva tarjeta debe desplazarse hasta el final de la lista con el gesto “swipe” de derecha a izquierda, hasta llegar al icono de una tarjeta que contiene el signo “+” (figura 17). Pulsando sobre él se abrirá el formulario para introducir los datos de la nueva tarjeta.

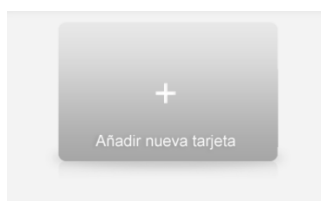


Figura 17. Icono de tarjeta para abrir el formulario “Nueva Tarjeta”

4.4.6 Sección Explora

Una de las funcionalidades que ofrece la aplicación es realizar el check in mediante el uso de la geolocalización.

Desde la sección “Explora” (figura 18) el usuario puede ver una lista ordenada de los establecimientos que se encuentran más cercanos a su ubicación actual.

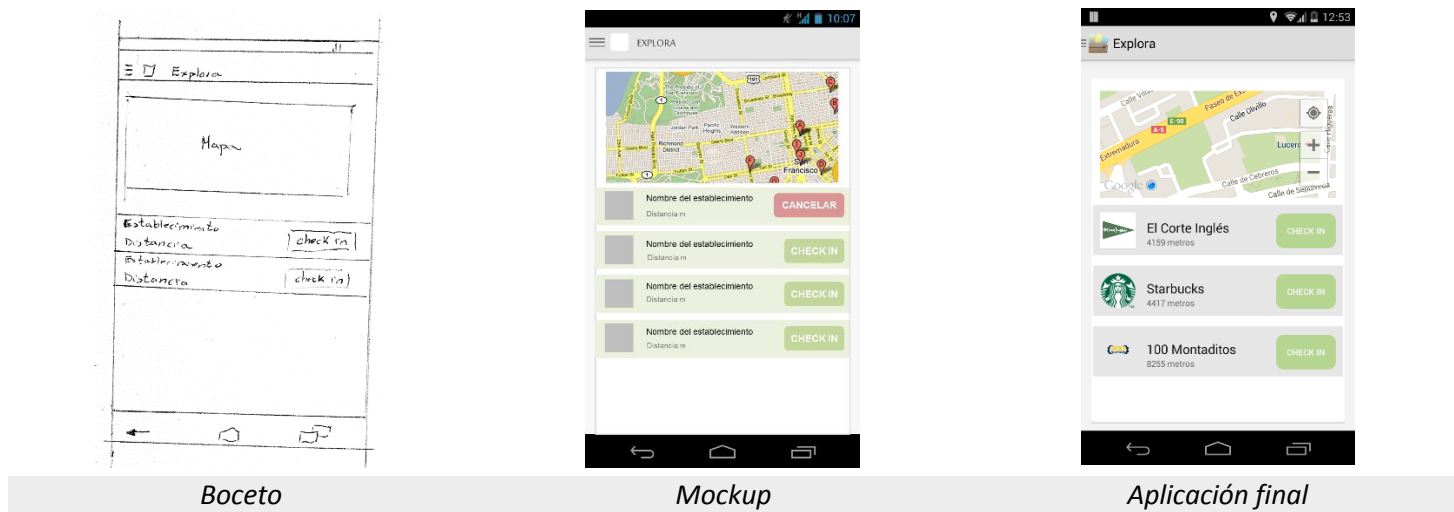


Figura 18. Proceso de diseño de la sección Explora

Junto con la lista se muestra un mapa (figura 19) que indica donde está el usuario y los establecimientos.

El mapa será especialmente útil si el usuario se encuentra en una ciudad que no conoce y quiere utilizar la modalidad de pago “check in”, ya que podrá ver mediante marcadores las ubicaciones de los establecimientos que la permiten.

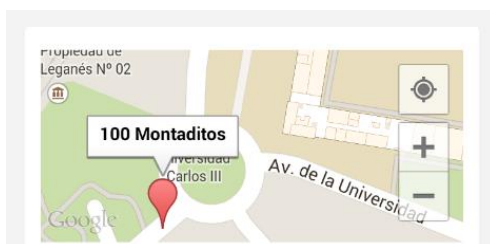


Figura 19. Marcador de un establecimiento cercano en la sección explora

En la lista se muestra el nombre del establecimiento, la distancia a la que se encuentra en metros y un botón que al ser pulsado abre la pantalla de check in, donde el usuario configurará los datos del mismo.

Además, en la lista también se muestra el logotipo del establecimiento de modo que el usuario podrá reconocerlo más fácilmente.

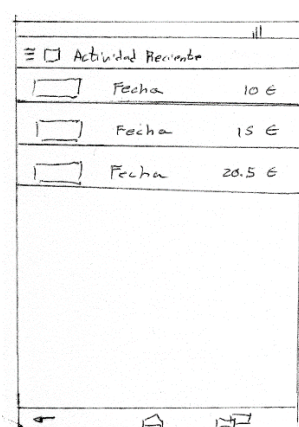
4.4.7 Sección Actividad Reciente

Desde esta sección el usuario puede ver todas las transacciones realizadas desde la aplicación.

En cada elemento de la lista se muestra la fecha y el importe por el que se realizó el pago. Junto con estos datos se muestra también una imagen de la tarjeta, de modo que el usuario puede identificar fácilmente con cual realizó cada una de sus compras.

Al pulsar sobre una de las transacciones se abre un recibo detallado de la misma, donde se puede ver una información más detallada (como el establecimiento y si se usaron o no ofertas). El recibo se analizará en detalle en el apartado 4.1.12.

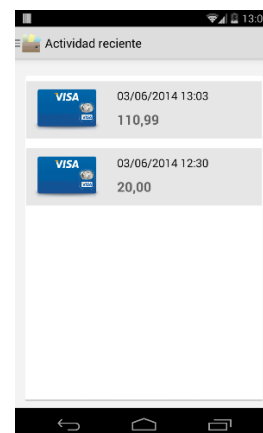
La figura 20 muestra el proceso de diseño.



Boceto



Mockup



Aplicación final

Figura 20. Proceso de diseño de la sección Actividad Reciente

4.4.8 Sección Cupones y descuentos

Otra de las funcionalidades que ofrece mWallet a los usuarios es la posibilidad de gestionar las ofertas y descuentos de los establecimientos (figura 21).

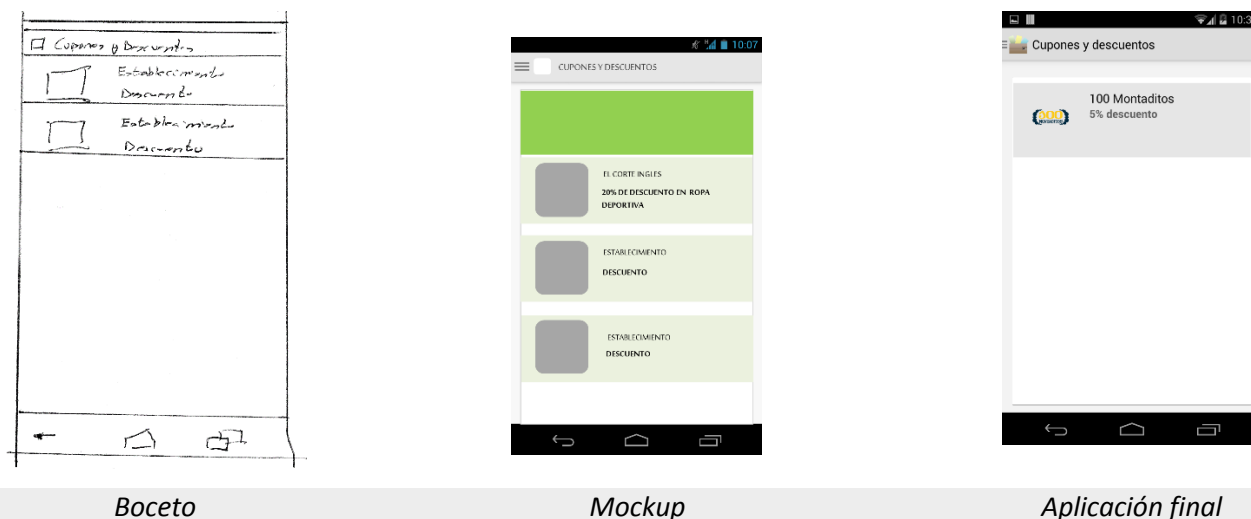


Figura 21. Proceso de diseño de la sección Cupones y Descuentos

Desde esta sección, el usuario puede ver un listado con las ofertas que tiene disponibles. El diseño realizado mantiene la misma estructura que el resto de la aplicación. Se muestra en cada elemento de la lista el nombre del establecimiento y el descuento que se ofrece.

Al igual que en el resto de las listas, se muestra junto con esta información una imagen con el logotipo del establecimiento de modo que el usuario pueda reconocerlo de manera más sencilla.

Una vez el usuario haya aplicado una oferta en uno de sus pagos, ésta desaparecerá de la lista.

4.4.9 Sección Comparte los gastos

Esta funcionalidad fue incluida en la aplicación estando ya avanzado el desarrollo de la misma, debido a esto solo se realizó un boceto a partir del cual se diseñó la interfaz final (figura 22).



Figura 22. Proceso de diseño de la sección Comparte los Gastos

Desde esta sección un usuario puede generar un código QR que es leído por otros usuarios de mWallet para realizar una transferencia a la tarjeta especificada por el primero. Esta interacción es explicada con detalle en el apartado 4.2 de este capítulo.

En la pantalla se selecciona la tarjeta a la que se quiere que se le haga la transferencia y se indicará el importe de la misma. Al pulsar sobre el botón “Generar Código” se mostrará una pantalla con el código QR que debe ser leído por el resto de usuarios que participen en la transacción compartiendo el gasto.

4.4.10 Sección Configuración

La última sección mostrada en el menú lateral es la de configuración (figura 23). En esta pantalla el usuario puede modificar los datos de su cuenta: nombre, apellidos y email.

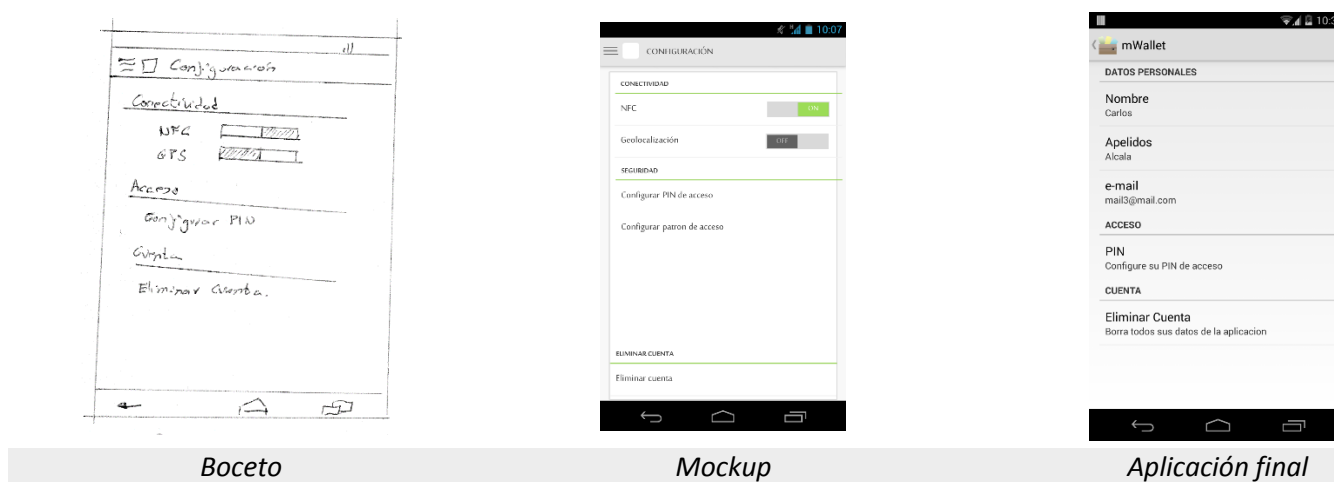


Figura 23. Proceso de diseño de la sección Configuración

Además, puede también configurar el PIN de acceso nuevamente o eliminar su cuenta, borrando así toda su información del sistema.

El diseño final de la pantalla tiene diferencias significativas con el boceto y el mockup que se elaboró en un primer momento.

En el mockup se permitía que el usuario pudiese activar o desactivar tanto la conectividad NFC como la geolocalización, sin embargo, por motivos de seguridad, esta funcionalidad no puede ser implementada en Android y esos parámetros solo pueden ser cambiados desde los ajustes del sistema.

4.4.11 Proceso de pago

En mWallet, como se vio en el capítulo 3, existen dos modalidades de pago: check in y cash desk, pudiéndose realizar ambas con NFC o QR indistintamente.

En este apartado se van a explicar las interfaces para cada una de las dos tecnologías.

Check In

Cuando el usuario lee un código QR con la aplicación o pulsa sobre el check in desde la sección “Explora”, se le muestra una pantalla que detalla los datos del mismo (figura 24).

La única diferencia entre el check in a través de QR y geolocalización es que en el caso del primero, la tarjeta ya se encuentra seleccionada, mientras que en el segundo se muestra un desplegable desde donde el usuario la escoge.

Además de la tarjeta seleccionada se muestra el nombre del establecimiento y los datos personales del usuario, que se asegura de este modo que toda la información que va a enviar es correcta.

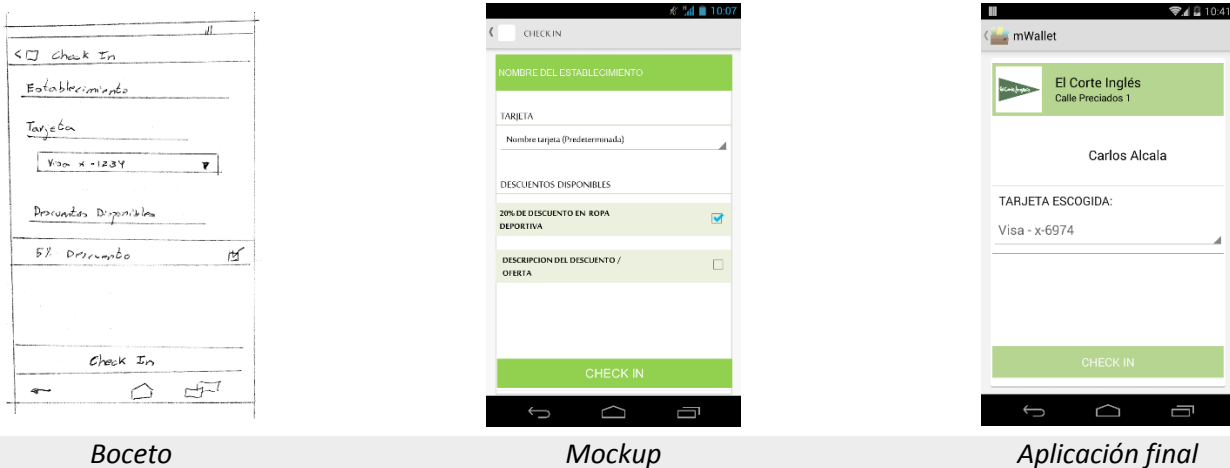


Figura 24. Proceso de diseño del formulario de check in

Tras confirmar el check in se muestra una pantalla informando al usuario que se ha realizado correctamente y que puede cancelarlo si lo desea desde la actividad reciente.

Además, siguiendo la heurística [37] “Visibilidad del estado del sistema”, desde la “Cartera” se indicará al usuario que tiene un check in activo, como muestra la figura 25.

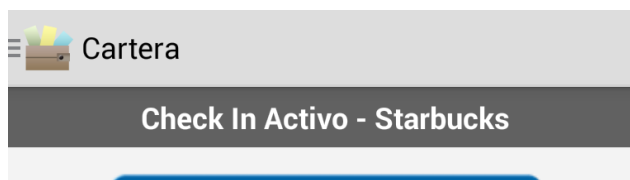


Figura 25. Barra indicadora de check in activo incluida en el diseño final de la aplicación

Cash Desk

En el caso del cash desk, una vez que se ha seleccionado una tarjeta desde la “Cartera”, se procede a realizar el pago.

Si se utiliza la tecnología NFC se muestra una primera pantalla (figura 26) que indica que se ha detectado la conexión NFC y se está procesando el pago.

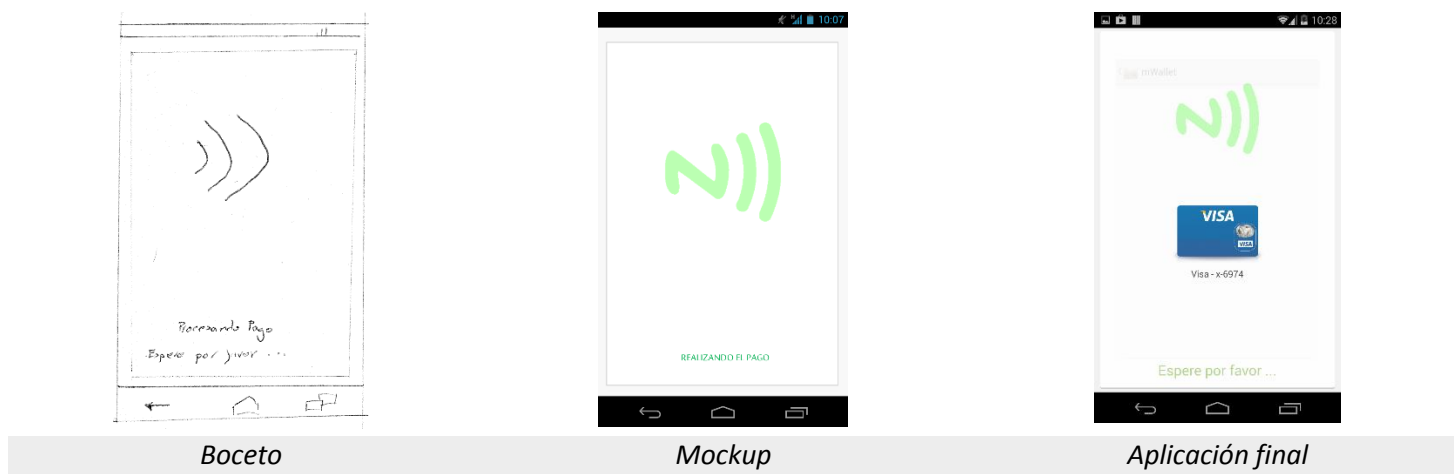


Figura 26. Proceso de diseño del pago a través de cash desk

En el diseño final se decide incluir también la información de la tarjeta para que el usuario pueda verificar que todo es correcto.

Si hay alguna oferta disponible aparece una pantalla que permitirá aplicarla o no al pago.

4.4.12 Recibo

Tras realizar un pago o al acceder a la actividad reciente, el usuario puede ver un recibo donde se detalla toda la información del mismo (figura 27).

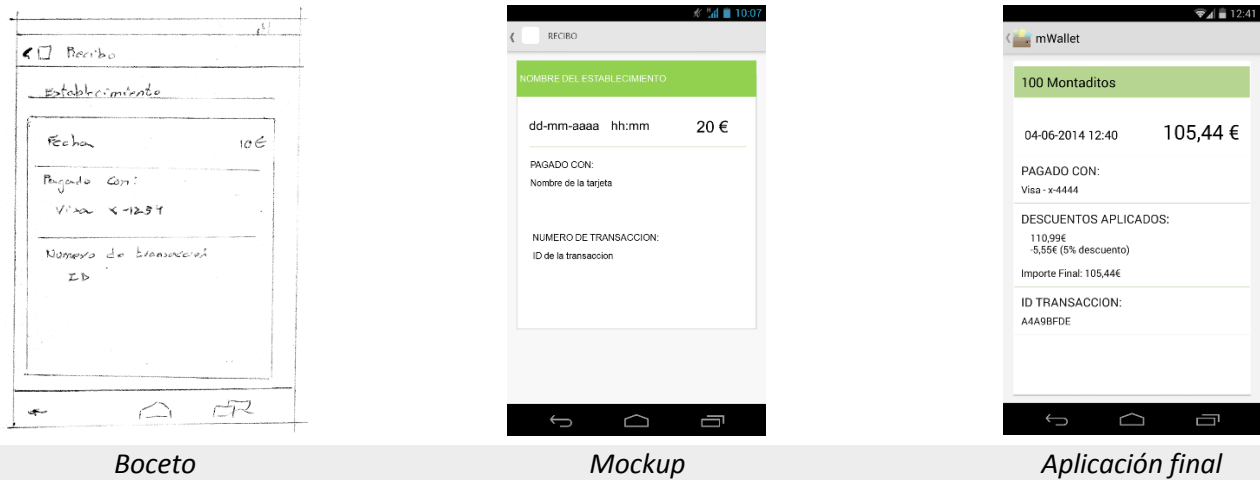


Figura 27. Proceso de diseño del Recibo

En el recibo se especifica el nombre del establecimiento, la fecha y hora en que se completó el pago y el precio final.

También se indica la tarjeta utilizada, que, al igual que ocurría en la “Cartera”, solo muestra los últimos 4 dígitos por motivos regulatorios de seguridad [34].

Además, si el usuario, al realizar su compra, utiliza uno de los descuentos aplicables, se muestra de forma detallada su desglose.

Cada operación se identifica con un ID de transacción que también es mostrado al usuario.

En esta pantalla se ha incluido una animación que simula la creación de un recibo en cualquier caja registradora. De esta forma se aprovecha el *user experience* y la aplicación resulta más familiar.

4.5. Detalles de Implementación

4.5.1 Arquitectura del sistema

El sistema desarrollado sigue una arquitectura cliente - servidor:

- La aplicación móvil tomará el rol de **cliente**, y enviará solicitudes al servidor para obtener información sobre las tarjetas de crédito, establecimientos cercanos u ofertas entre otros.

Además, cuando el usuario realiza una transacción (cash desk o check in), la información de la misma es enviada al servidor para que pueda ser procesada.

- El **servidor** está formado por un servidor web tomcat en su versión 7.0 y una base de datos relacional MySQL. El servidor web contiene dos aplicaciones, un prototipo básico de aplicación que permite al establecimiento completar un check in activo y un conjunto de servicios web que se encargan de realizar la comunicación entre el cliente y la base de datos. Ambas están implementadas en el lenguaje de programación java.

La comunicación entre cliente y servidor se realiza mediante el protocolo de intercambio de mensajes SOAP (Simple Object Access Protocol).

Además, la aplicación también deberá procesar la información recibida de los tags NFC y los códigos QR, mediante los cuales se realizan los modelos de interacción.

La figura 28 representa los distintos componentes del sistema descritos anteriormente:

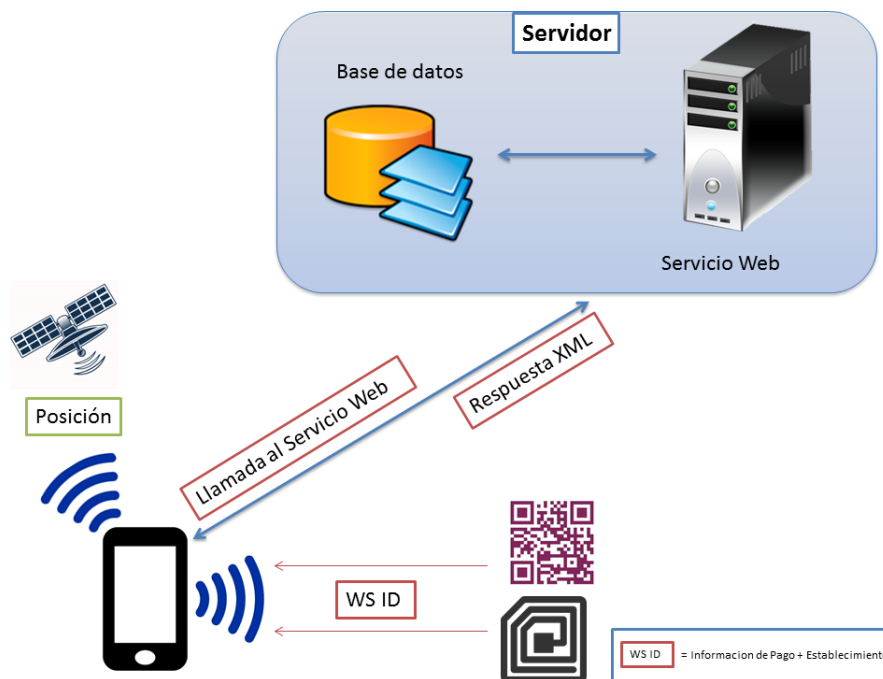


Figura 28. Arquitectura de mWallet

4.5.2 Base de datos

El sistema desarrollado cuenta con una base de datos relacional MySQL que almacena toda la información necesaria para la gestión de: usuarios, tarjetas, establecimientos y transacciones.

A continuación se muestra el diseño de la base de datos utilizada (figura 29).

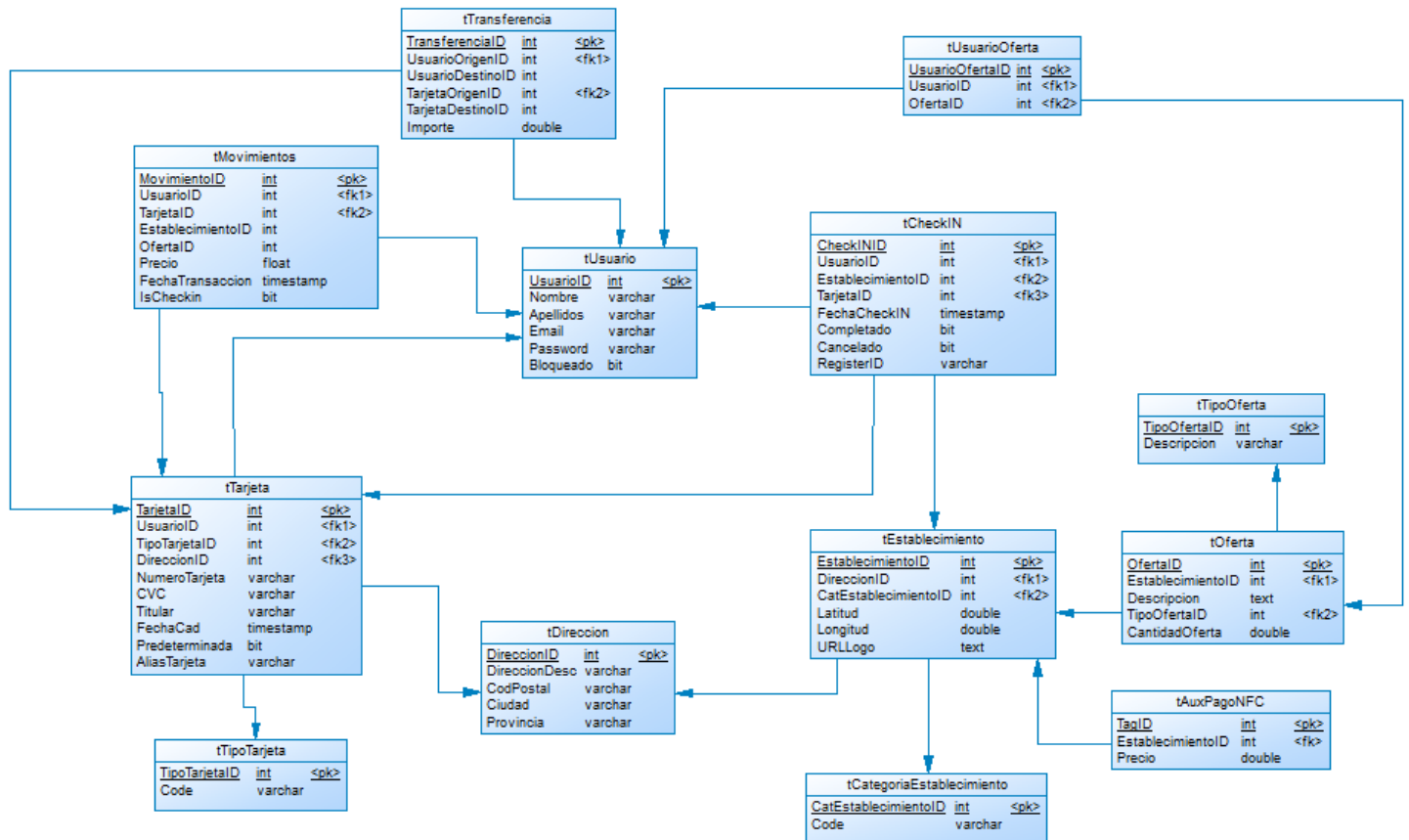


Figura 29. Diseño relacional de la base de datos MySQL

Existen algunas tareas que implican el uso de consultas complejas y que se van a repetir a menudo en el sistema. Para ello se han creado **procedimientos almacenados** que permiten simplificar el código java del servicio web liberándole de la programación SQL, quedando el código en una llamada al procedimiento.

Además, estos procedimientos se ejecutan directamente en el motor de la base de datos lo que permite mejorar el rendimiento con respecto a realizar varias consultas desde el servicio web.

pNuevaTarjeta Load

Este procedimiento es utilizado cuando el usuario añade una nueva tarjeta en la aplicación.

Al añadir una nueva tarjeta el usuario introduce los siguientes campos:

- Titular de la tarjeta
- Número de la tarjeta

- Tipo de tarjeta
- Fecha de caducidad
- Código CVC
- Dirección
- Código Postal
- Provincia
- Ciudad

Junto con estos datos el procedimiento también recibe el identificador del usuario y el alias de la tarjeta (que es generado desde la aplicación).

Este procedimiento debe incluir datos en las tablas tTarjeta y tDireccion, esta última no debe almacenar dos veces una misma dirección, por tanto el procedimiento debe comprobar esto antes de insertar.

El algoritmo que sigue el procedimiento es el siguiente:

1. Comprueba si la dirección recibida como parámetro existe ya en la tabla tDireccion
 - a. Si existe
 - i. Obtiene el identificador asociado a esa dirección
 - ii. Inserta los datos de la nueva tarjeta en tTarjeta usando como identificador de la dirección el obtenido anteriormente.
 - b. Si no existe
 - i. Inserta los datos de la dirección en tDireccion
 - ii. Obtiene el identificador de la dirección que ha sido insertada en el paso anterior
 - iii. Inserta los datos de la tarjeta en tTarjeta

pEstablecerPredeterminada

Este procedimiento es llamado cuando un usuario establece como predeterminada una tarjeta de su lista.

El procedimiento debe actualizar el campo “Predeterminada” de la tabla tTarjeta para poner el bit a 1 de la tarjeta que se desea predeterminar y poner a 0 el de la anteriormente establecida como predeterminada.

El procedimiento recibe como parámetros:

- Identificador del usuario
- Identificador de la tarjeta a predeterminar

El algoritmo que sigue el procedimiento es el siguiente:

1. Pone a 0 el bit del campo Predeterminada de las tarjetas del usuario que cuyo identificador es distinto del de la tarjeta que se va a predeterminar.
2. Pone a 1 el bit del campo Predeterminada de las tarjetas del usuario que cuyo identificador es igual al recibido como parámetro.

En el anexo I - Diseño de la base de datos, se detallan los aspectos más relevantes de cada una de las tablas.

4.5.3 Frameworks utilizados

4.5.3.1 Android

La aplicación mWallet ha sido desarrollada para el sistema operativo Android.

En la implementación se ha utilizado la API 14, que se corresponde con la versión 4.0 de dicho sistema operativo. Esto se debe a que la aplicación utiliza librerías NFC que solo están disponibles a partir de esta versión.

Además, en la aplicación, también se incluye el Navigation Drawer explicado anteriormente, junto con el action bar. Estas dos formas de navegación se encuentran únicamente disponibles a partir del nivel 14 de la API de Android.

Por último, el mapa utilizado en la sección “Explora” también requiere de este nivel de API.

La limitación que supone utilizar la aplicación desde la versión 4.0 dejando fuera a dispositivos con versiones anteriores de Android, se considera asumible ya que la mayoría de los terminales con versiones anteriores a la 4.0 no dispone de un módulo NFC.

La aplicación se compone de una actividad principal (*Activity_mWallet_Main.java*) que tiene un layout compuesto por fragments.

Los fragments [25] son porciones de interfaz de usuario que pueden ser añadidas o eliminadas de forma dinámica de la actividad. Estos siempre se encuentran embebidos dentro de la actividad. La figura 30 muestra el comportamiento de los fragments en una aplicación.

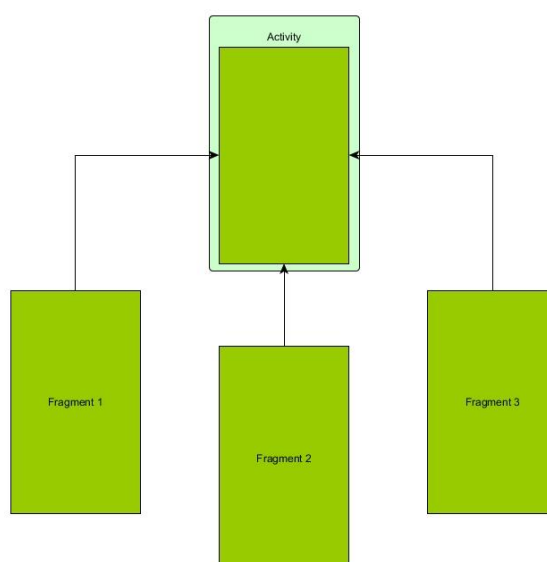


Figura 30. Diagrama explicativo del funcionamiento de los fragments

La gestión de los fragments se realiza a través del menú lateral (Navigation Drawer). Cuando se pulsa sobre una de las secciones de la aplicación se carga la interfaz correspondiente al fragment asociado.

Almacenamiento de la información

mWallet almacena en el dispositivo móvil información del usuario: nombre, apellidos, PIN y un bit de control para saber si está o no registrado. Lo hace en las preferencias compartidas [26]. La imagen de perfil se guarda en el almacenamiento externo del dispositivo.

Las preferencias de usuario tienen 3 modos de acceso:

- **MODE_WORLD_WRITABLE:** Todas las aplicaciones instaladas pueden acceder y modificar las preferencias.
- **MODE_WORLD_READABLE:** Todas las aplicaciones instaladas pueden leer las preferencias pero solo pueden ser modificadas por la aplicación que las crea.
- **MODE_PRIVATE:** Solo la aplicación que las crea puede acceder a ellas.

En el prototipo desarrollado se usará el **MODE_PRIVATE** para poder asegurar el sandboxing (aislamiento) de la aplicación respecto a las demás aplicaciones, ya que se guarda información importante como el PIN del usuario.

```
settings = getSharedPreferences("com.uc3m.mwallet_preferences", MODE_PRIVATE);
```

...

```
Editor editor = settings.edit();  
editor.putString("PIN", pin);  
editor.commit();
```

Además de las preferencias de usuario, también se guarda información en el almacenamiento externo [26] (imagen de perfil). Estos datos, a diferencia de las preferencias compartidas, pueden ser accedidos por todas las aplicaciones y también por el usuario cuando tenga acceso físico al dispositivo.

En el código se especifica el directorio donde se debe almacenar la imagen, y en caso de que no exista se creará.

```
File mediaStorageDir = new File(Environment.getExternalStoragePublicDirectory(  
    Environment.DIRECTORY_PICTURES), "mWallet");
```

4.5.3.2 SOAP y GCM (Google Cloud Messaging)

SOAP (Simple Object Access Protocol)

Las comunicaciones iniciadas por la aplicación utilizan el protocolo de comunicación SOAP [27]. Este protocolo es independiente del lenguaje, lo que permite la intercomunicación entre sistemas que utilicen distintas tecnologías y distintos lenguajes de programación. Todos los mensajes SOAP siguen un esquema XML.

En el prototipo desarrollado todas las peticiones al servidor son realizadas a través de una *tarea asíncrona* [28].

En Android, todas las actividades tienen lugar en el mismo hilo de ejecución, denominado GUI Thread. Aquí se controla la interfaz y se realizan todas las operaciones.

Por ello, si se realiza una operación costosa sobre este hilo, se bloquearía la aplicación, incluyendo la interfaz.

En **mWallet**, la mayoría de las operaciones requieren realizar una conexión con el servidor que la procesa realizando una consulta sobre la base de datos y respondiendo a la aplicación.

Esta operación es costosa ya que requiere realizar una conexión a través de internet y esperar a que se reciba la respuesta, y por tanto, si se ejecuta sobre el hilo principal de ejecución, bloquearía la aplicación. Por ello, se han utilizado tareas asíncronas que se ejecutan en un hilo distinto al principal.

Para implementar una tarea asíncrona se crea una nueva clase dentro de la actividad donde se quiera utilizar y se definirán 3 métodos:

- `onPreExecute`: donde se preparan los datos que son enviados al servidor.
- `doInBackground`: en este paso se realiza la llamada al servicio web a través de un mensaje SOAP
- `onPostExecute`: Se ejecuta una vez ha terminado la tarea. Aquí se procesará la respuesta adaptando la interfaz a la misma.

En el prototipo desarrollado todas las peticiones al servidor son realizadas a través de una tarea asíncrona.

La aplicación utiliza la librería KSOAP para realizar las peticiones de datos al servidor y obtener la respuesta a las mismas.

KSOAP es una librería eficiente y muy utilizada para generar mensajes SOAP en Android. El proceso para generar los mensajes sigue siempre el mismo esquema. En la clase interna creada en la actividad se definen cuatro atributos que identifican al servidor, al servicio web y al método que se quiere llamar.

En el método `onPreExecute`, se obtienen los valores que se envían al servicio web. Una vez preparado todo, se invoca el método `doInBackground`, que prepara el mensaje SOAP.

```
SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);
```

Una vez creado el objeto SOAP se le pasan los parámetros del servicio web. Se muestra como ejemplo la petición de los establecimientos cercanos, donde se envían latitud y longitud

```
request.addProperty("latitud", String.valueOf(latitude));  
request.addProperty("longitud", String.valueOf(longitude));
```

A continuación se creará el envelope del mensaje SOAP que contendrá la información anterior y se enviará la solicitud al servidor, quedando a la espera de la respuesta por parte del mismo.

Por su parte, en el **lado servidor**, los servicios web obtienen los parámetros enviados por la aplicación, realizan la consulta correspondiente a la base de datos y generan una respuesta, a la que se le da un formato XML, que es procesado por la aplicación.

A continuación, se muestra como ejemplo la respuesta del servidor a la petición de la lista de tarjetas de un usuario:

```
<ListaTarjetas>
  <Tarjeta>
    <TarjetaID></TarjetaID>
    <Tipo></Tipo>
    <Numero></Numero>
    <FCad></FCad>
    <Predeterminada></Predeterminada>
  </Tarjeta>
</ListaTarjetas>
```

En la respuesta habrá tantos elementos “Tarjeta” como tarjetas haya añadido el usuario. Este mensaje es embebido en uno SOAP, que es recibido por la aplicación.

Al llegar a la aplicación, a través de los métodos proporcionados por la librería KSOAP, se extrae esta respuesta y mediante la clase XMLParse implementada se obtiene toda la información incluida dentro de las etiquetas XML.

De este modo, cuando se recibe un mensaje del servidor, lo primero que se realiza es llamar al método correspondiente de la clase de utilidad *XMLParse.java*. En el caso de la solicitud de establecimientos cercanos, se devolverá un ArrayList de objetos de tipo Establecimiento.

```
establecimientosList = XMLParse.obtenerEstablecimientos(xmlRespuesta);
```

Una vez procesada la respuesta se invoca el método `onPostExecute` y se aplican los cambios necesarios sobre la actividad. Siguiendo con el ejemplo de los establecimientos, se generará la lista de los mismos a partir de `arraylist`.

```
AdapterExploraList adapter = new
AdapterExploraList(getActivity().getApplicationContext(),
establecimientosList);

lista.setAdapter(adapter);
```

GCM (Google Cloud Messaging)

Cuando un usuario realiza un check in, su información es enviada al establecimiento de tal modo que en el momento del pago solo tiene que indicar su nombre.

En el establecimiento comprueban que realmente existe un check in con ese nombre y procesan el pago. En este momento, el cliente debe ser informado de que la transacción se ha realizado correctamente. En mWallet esto se hace a través de una notificación push.

A diferencia de la comunicación mediante SOAP, en la que es el cliente el que la inicia, en las notificaciones push es el servidor el que debe empezar la comunicación.

A partir de la versión 2.2 de Android, Google introdujo el servicio Google Cloud Messaging (GCM) [29]. Para este servicio google utiliza un servidor intermedio entre la aplicación y el servidor que es el encargado de recibir las notificaciones y reenviarlas al dispositivo correspondiente.

El funcionamiento de GCM se describe en la figura 31:

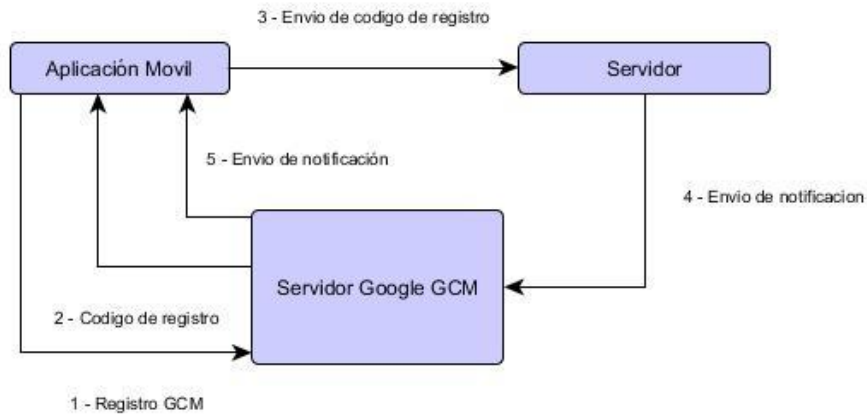


Figura 31. Diagrama de flujo del protocolo de comunicación GCM

1. La aplicación envía una petición de registro al servidor GCM
2. El servidor responde a la petición con un código de registro que debe ser almacenado tanto por la aplicación como por el servidor
3. La aplicación envía el código de registro al servidor para que sea almacenado.
4. El servidor envía un mensaje, utilizando el código de registro, al servidor GCM.
5. El servidor GCM reenvía el mensaje al dispositivo, donde se gestionará. Por ejemplo, mostrando una notificación.

En mWallet el proceso de registro se realiza cuando el usuario confirma el check in. En este momento, la aplicación recibe el código de registro, que es enviado en el mensaje SOAP junto con el resto de datos del check in.

Al mismo tiempo, durante la ejecución de mWallet existe un servicio activo que está esperando los mensajes del servidor GCM. Cuando se completa el check in, el servicio recibe un mensaje que contiene el identificador del establecimiento, el identificador de la tarjeta y el importe. Con estos datos la aplicación crea una notificación que se muestra al usuario.

Al pulsar sobre la notificación aparece el recibo completo de la operación.

Para poder simular un pago a través de check in se ha desarrollado un prototipo básico de aplicación web que sería utilizado en los establecimientos para que el personal del mismo complete el pago cuando el usuario le indique su nombre.

Esta aplicación web se compone de dos páginas JSP: un formulario para indicar el establecimiento y un listado de los check in activos.

Además, también se cuenta con dos servlets que obtienen la lista de los check in de base de datos y permiten completar el pago enviando un mensaje a la aplicación.

El servlet que permite obtener la lista de check in recibe de la página JSP el identificador del establecimiento. A partir de este dato se realiza una consulta a la base de datos, similar a la vista en el caso de los servicios web, que devuelve todos los check in activos en ese establecimiento.

Esta lista es pasada como parámetro a una página jsp que se encargará de mostrarlo.

Desde esa página jsp, el personal del establecimiento puede marcar un check in como completado. En ese momento se llama el segundo servlet, donde se actualiza el estado del check in en base de datos y pasa a “completado” enviando posteriormente una notificación al dispositivo móvil.

Este último paso hace uso de la comunicación GCM. Para poder acceder a los métodos que permiten GCM es necesario incluir la librería correspondiente en la aplicación (*gcm-server.jar*). A través del código de registro, que se encuentra almacenado en la base de datos, se crea un mensaje y se envía al servidor de GCM.

4.5.3.3 Google Maps V2

En la sección Explora se sitúa un mapa en la parte superior de la pantalla donde se muestra la localización actual del usuario así como la de los establecimientos cercanos que aceptan el check in.

Para incluir este mapa se utiliza la versión 2 de la API de Google Maps para Android [30]. Esto se hace a través de un fragment específico (MapFragment) para este contenido.

```
<fragment
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:layout_margin="10dp"
    class="com.google.android.gms.maps.SupportMapFragment"
    android:name="com.google.android.gms.maps.MapFragment"/>
```

Este fragment se incluye dentro del layout de la propia sección explora.

Al utilizar los mapas es necesario dar permisos a la aplicación. Estos se especifican en el AndroidManifest.

Los permisos que necesitan los mapas de google para su funcionamiento son:

- Acceso a internet
 - "android.permission.INTERNET"
- Conocer el estado de la red
 - "android.permission.ACCESS_NETWORK_STATE"
- Acceder al almacenamiento externo para leer la cache de mapas
 - "android.permission.WRITE_EXTERNAL_STORAGE"
- Uso de los servicios de google
 - "com.google.android.providers.gsf.permission.READ_GSERVICES"

- Acceso a la localización
 - "android.permission.ACCESS_FINE_LOCATION"
 - "android.permission.ACCESS_COARSE_LOCATION"

estos últimos no son necesarios si bien se recomiendan.

Desde la sección Explora, la aplicación envía una solicitud al servidor para obtener la lista de los establecimientos cercanos. En esta petición son pasados como parametros la latitud y longitud (posición actual del usuario), a partir de las cuales el servicio web (*obtenerEstablecimientosCheckin*) calcula, utilizando la ley esférica del coseno, la distancia a los establecimientos, cuya fórmula es la siguiente:

$$c = \text{acos}(\sin(\text{lat1}) * \sin(\text{lat2}) + \cos(\text{lat1}) * \cos(\text{lat2}) * \cos(\text{long2} - \text{long1}))$$

$$\text{Distancia} = \text{Radio} * c \text{ siendo } R \text{ el radio de la tierra en metros (6371 km)}$$

La consulta SQL que utiliza esta ecuación es la siguiente:

```
SELECT
    (6371 * acos (cos ( radians("+Double.parseDouble(latitud)+") )
    * cos( radians( Establecimiento.Latitud ) ) * cos( radians(
    Establecimiento.Longitud ) -
    radians("+Double.parseDouble(longitud)+") ) + sin (
    radians("+Double.parseDouble(latitud)+") ) * sin( radians(
    Establecimiento.latitud ) ) ) ) AS distancia
FROM
    tEstablecimiento Establecimiento

INNER JOIN
    tDireccion Direccion
    ON Direccion.DireccionID = Establecimiento.DireccionID

ORDER BY distancia
```

Una vez recibida la información de los establecimientos, desde el código, se incluyen de forma dinámica marcadores en las ubicaciones de los mismos.

Ya que el usuario puede estar en movimiento, se utiliza un listener para que en caso de que se produzca un cambio de localización, se actualice el mapa y la lista de establecimientos cercanos. Además, se solicita un refresco del mapa cada 30 segundos.

```
locationManager.requestLocationUpdates(
    LocationManager.NETWORK_PROVIDER, 30000, 0, locListener);
```

4.5.3.4 Librería ZXing: Lectura de códigos QR

La aplicación mWallet es capaz de capturar códigos QR y procesar la información contenida en los mismos para realizar las transacciones. Para que las operaciones puedan ser realizadas de forma correcta, es necesario que el contenido del QR siga un formato específico.

Se han definido 3 tipos de operaciones a realizar con los códigos QR:

- Check in
- Transferencias
- Cash Desk

Cada una de estas tiene asociado un código de operación y los campos específicos para poder realizar la transacción. A continuación, se detallará cuál es el formato que debe seguir el contenido del código QR en cada una de las operaciones.

Check In QR

En esta modalidad de pago, el usuario, al leer el código QR, obtiene la información relativa al establecimiento.

El formato que debe seguir el contenido del código QR para esta operación es el siguiente:

```
<mWalletQR>
  <IDOp>100</IDOp>
  <Establecimiento>
    <EstablecimientoID></EstablecimientoID>
    <Nombre></Nombre>
    <Direccion></Direccion>
    <Logo></Logo>
  </Establecimiento>
</mWalletQR>
```

Transferencia (compartir gastos)

En el caso de las transferencias, el usuario, al leer el código QR, obtiene el importe de la misma así como los datos de la tarjeta a la que se va a realizar. El formato que debe seguir el contenido del código QR para esta operación es el siguiente:

```
<mWalletQR>
  <IDOp>101</IDOp>
  <Transferencia>
    <UsuarioID></UsuaioID>
    <TarjetaID></TarjetaID>
  </Transferencia>
</mWalletQR>
```

```
<Importe></Importe>

</Transferencia>

</mWalletQR>
```

Siendo UsuarioID y TarjetaID datos relacionados con el usuario al que se le realiza la transferencia.

Cash Desk QR

En esta modalidad de pago, el usuario, al leer el código QR, obtiene los datos del establecimiento y el importe por el que se va a realizar el pago.

El formato que debe seguir el contenido del código QR es el siguiente:

```
<mWalletQR>

  <IDOp>102</IDOp>

  <CashDesk>

    <EstablecimientoID></EstablecimientoID>

    <Importe></Importe>

  </CashDesk>

</mWalletQR>
```

Para implementar la lectura de los códigos QR se ha decidido utilizar la librería *ZXing* [31]. Se trata de un proyecto de código abierto que proporciona métodos para gestionar y procesar los QR.

En primer lugar y tras importar la librería al proyecto de la aplicación, se realiza un análisis de la misma ya que contiene una gran cantidad de clases que no son necesarias para la interacción que se busca y harían que la aplicación fuese más pesada.

Solo se utilizarán los métodos relativos a la lectura de códigos y a la generación de los mismos.

Al añadir esta funcionalidad, ha sido necesario dar el permiso de acceso a la cámara, que también es utilizado durante el proceso de registro para realizar la fotografía del usuario.

```
<uses-permission android:name="android.permission.CAMERA" />
```

No obstante, ha sido necesario modificar algunos métodos de la librería para que se adaptasen a la interacción de los pagos. En particular, ha sido necesario añadir a los extras del intent el identificador de la tarjeta, ya que la lectura de un código QR siempre se realiza asociada a una tarjeta de crédito del usuario.

```
Intent intent = new Intent("com.google.zxing.client.android.SCAN");
intent.putExtra("SCAN_MODE", "QR_CODE_MODE");
intent.putExtra("SAVE_HISTORY", false);
intent.putExtra("tarjeta", nombreTarjeta);
intent.putExtra("IDTarjeta", TarjetaID);
startActivityForResult(intent, 0);
```

4.5.3.5 NFC

La aplicación utiliza la comunicación NFC para permitir a los usuarios realizar los pagos. Para ello se incluye en el código de la aplicación el paquete *android.nfc*, mediante el cual será posible acceder a los métodos que se usan para gestionar la lectura de los tags NFC.

Si el módulo NFC se encuentra activo en el dispositivo, cuando se detecta un tag NFC, se procede a su lectura. Esta característica supone un obstáculo para la aplicación desarrollada ya que se quiere que únicamente el tag sea leído cuando el usuario tenga seleccionada una tarjeta desde la “Cartera”.

Como se ha comentado en el apartado anterior la aplicación se compone de una actividad principal sobre la que se cargan los fragments de cada sección.

La gestión de la detección de un tag NFC se hace dentro de la actividad principal. Es necesario gestionar por tanto aquí si el fragment cargado en la actividad se corresponde con el de la “Cartera”, en caso contrario se deberá mostrar un mensaje indicando al usuario que no ha escogido una tarjeta válida.

Cuando el tag es leído, se manda un nuevo intent a la aplicación, que es gestionado desde el método *OnNewIntent*. Desde aquí se obtiene el ID del tag, controlando previamente que esté seleccionada una tarjeta válida. En caso de que todo sea correcto se comenzará el proceso de pago.

4.5.3.6 Apache Axis2: Servicios Web

Los servicios web son ejecutados a través del motor de servicios web/SOAP/WSDL Axis2. Se han implementado tres servicios web que agrupan todas las operaciones necesarias para el funcionamiento de la aplicación:

- GestionPagos.java
Contiene métodos que permiten insertar nuevos pagos, realizar check in y obtener información sobre las ofertas.
- GestionTarjetas.java
Proporciona métodos para añadir una nueva tarjeta, listar las tarjetas de un usuario y ver los movimientos realizados por las mismas.
- PerfilUsuario.java
Proporciona los métodos para el registro, borrado y obtención de los datos de un usuario.

Las consultas realizadas a la base de datos se hacen utilizando la API de java JDBC, que se encuentra en el paquete *java.sql*. Éstas se hacen a través de llamadas directas a la base de datos. Para poder establecer la conexión con la base de datos, es necesario también incluir el driver de MySQL en el classpath de la aplicación.

Los pasos que realizan los servicios web para realizar una consulta son los siguientes:

- Obtener el driver de conexión de la base de datos MySQL
`Class.forName("com.mysql.jdbc.Driver");`
- Establecer la conexión con la base de datos indicando el usuario y contraseña a partir del driver obtenido en el paso anterior.
- Crear la sentencia SQL utilizando objetos PreparedStatement
`String sql = "SELECT COUNT(*) AS Numero FROM tUsuario WHERE Email=\""+email+"\"";`
`PreparedStatement statement = con.prepareStatement(sql);`
- Ejecutar las sentencias SQL guardando, en caso de que sea necesario, la respuesta a la consulta en un ResultSet
`ResultSet rs;`
`rs = statement.executeQuery();`
- Obtener la información del result set y generar una respuesta XML, que será enviada a la aplicación móvil.

5. Evaluación

5.1 Test de evaluación de la usabilidad

Con el objetivo de evaluar la facilidad de uso de la aplicación mWallet y comprobar la eficacia y eficiencia de los modelos de pago Cash Desk y Check In que se han desarrollado, se ha realizado una prueba piloto con usuarios.

En la prueba han participado 10 usuarios, dos de ellos de más de 50 años y el resto con edades comprendidas entre los 20 y 30 años. Todos ellos están familiarizados con el uso del Smartphone.

La evaluación se divide en cuatro fases:

- En primer lugar se realiza una explicación de la aplicación, detallando las dos modalidades de pago que se pueden realizar. Esta fase dura aproximadamente 10 minutos.
- A continuación se deja al usuario navegar libremente por la aplicación para que se familiarice con ésta y pueda dar una primera opinión sobre la misma.
- Una vez el usuario se siente cómodo con la aplicación se le pide que realice tres tareas en las que se utilizarán las modalidades de pago Cash Desk y Check In.
- Por último, se realiza una entrevista personal con cada usuario de unos 15 minutos en los que se pide que indique que modalidad de pago utilizaría más habitualmente y cuál es su opinión general sobre mWallet.
Al final de esta entrevista se entregará un cuestionario a los usuarios para que lo rellenen.

El cuestionario realizado en la última fase de la evaluación se basa en el estándar definido en [32] que permite evaluar la utilidad, la facilidad de uso, la facilidad de aprendizaje y la satisfacción. Se puede ver en el *Anexo II*.

Este estudio se centra en tres aspectos principales:

- Capacidad de entender la interfaz.
- Capacidad de navegar a través de la aplicación.
- Eficacia y eficiencia de los modelos de pago diseñados.

5.1.1 Experimento

Una vez explicado el funcionamiento básico de la aplicación, los usuarios dispusieron de un tiempo para navegar y familiarizarse con la misma.

Tras esta fase, se les propone realizar tres tareas que involucran las modalidades de pago diseñadas, para lo que se les situará en uno de los escenarios tipo vistos en el capítulo 3. Se le solicita que simulen un pago a través de:

- Cash Desk mediante NFC
- Check In mediante QR
- Check In mediante geolocalización.

La tabla 59 muestra los resultados obtenidos.

Tarea	Tiempo medio (s)	Tiempo estimado (s)	Media de errores	Errores previstos
Cash Desk con NFC	8	5	0	0
Check In con QR	11	10	0	0
Check In con Geolocalización	19	17	0	0

Tabla 59. Resultados

En la tabla 59 el tiempo medio y la media de errores de realización de cada tarea se comparan con el tiempo y errores cometidos por un usuario “experto” en la aplicación (tiempo estimado y errores previstos).

Como se puede ver, los tiempos no distan mucho del esperado, con lo que es posible deducir que la interfaz es efectiva y permite a los usuarios realizar las tareas de forma correcta, ya que no se cometen errores. Un aspecto importante a tener en cuenta y que puede ser la razón de estos resultados es que la interfaz es la misma para todos los métodos de pago, de modo que no solo no confunde al usuario sino que le ayuda a utilizar la aplicación con más facilidad.

Inmediatamente después de esta fase, se pide a los usuarios que rellenen un cuestionario [32] que permite evaluar distintos aspectos:

- Utilidad: se realizan preguntas relacionadas sobre si cubre las necesidades de los usuarios y les permite ser más efectivos al realizar sus tareas.
- Facilidad de uso: incluye preguntas sobre el esfuerzo necesario para utilizar la aplicación, el número de pasos para realizar las tareas y si son necesarias unas instrucciones para usarla.

- **Facilidad de aprendizaje:** se valora la facilidad y rapidez para adquirir las habilidades necesarias para usar la aplicación y si se recuerda cómo se emplea ésta de forma sencilla.
- **Satisfacción:** se realizan preguntas sobre si la aplicación funciona como se espera, es agradable y si le gustaría al usuario tenerla.

En este cuestionario se valora del 1 al 5 cada una de las categorías, siendo 5 “Totalmente de acuerdo” y 1 “Totalmente en desacuerdo”.

Se ha calculado la media de las evaluaciones para cada una de las categorías. Los resultados son mostrados en la tabla 60.

Utilidad	Facilidad de uso	Facilidad de aprendizaje	Satisfacción
4	3.7	4.2	4

Tabla 60. Valoración media de mWallet

El aspecto con una valoración más baja es el de “Facilidad de uso”. El resultado es bueno, pero el usuario tiene que emplear un pequeño esfuerzo al principio para aprender los distintos métodos de pago. Muchos de los usuarios indicaron que sería necesario incluir una pequeña ayuda para poder entender el uso de la aplicación de forma más sencilla y esto ha bajado la media de la valoración de este apartado.

No obstante, una vez aprendida, los usuarios pudieron realizar las tareas de forma correcta y rápidamente; “Es más cómodo sacar el móvil que tener que buscar la cartera dentro del bolso”, “usaría la aplicación de manera habitual, es muy rápida”.

Por último, se realiza una entrevista personal con cada uno de los usuarios donde se pide una opinión general sobre la aplicación, qué mejoras podría tener y qué modalidad de pago utilizarían de forma más frecuente.

En la tabla 61 se muestra el número de personas que usarían cada modalidad de pago.

Ninguna	Principalmente Cash Desk	Principalmente Check In	Ambas
1	3	2	4

Tabla 61. Preferencias en la utilización de los distintos medios de pago

Los usuarios que prefieren la modalidad de Check In indican que la utilizarían ya que es más rápida, “puedo hacer Check In desde la oficina y no tengo que preocuparme por nada más”.

Los usuarios que escogen el Cash Desk antes que el Check In lo hacen principalmente por razones de seguridad. Se sienten más cómodos usando la aplicación en el momento de la compra que enviando sus datos antes; “usaría el Check In solo en sitios que conociera muy bien”.

Cabe destacar que un usuario no utilizaría ninguna modalidad. Éste no suele realizar pagos con tarjeta de manera habitual y prefiere pagar en efectivo. Por esta razón, él no utilizaría la aplicación, aunque también indica que “la interfaz esta chula y se ve todo muy claro”

5.2 Consideraciones a la hora de diseñar un Mobile Wallet: Lecciones aprendidas

En este apartado se van a indicar algunas consideraciones que se deberían tomar en cuenta a la hora de diseñar mobile wallets y que se han podido extraer a partir de la evaluación con los usuarios.

Uno de los aspectos que más valoraron los usuarios fue el mapa incluido dentro de la sección “Explora”. Indicaron que gracias al mapa era posible encontrar más fácilmente donde estaban situados los establecimientos y poder así calcular una ruta a los mismos. También comentaron que sería muy útil si este mapa pudiese agrandarse, ya que en el prototipo no tiene un tamaño demasiado grande.

Al tratarse de una aplicación que trata datos muy sensibles es necesario que el usuario tenga en todo momento una sensación de seguridad. Esto se puede conseguir mediante mensajes informativos para confirmar algunas tareas.

Además de mostrar seguridad, la interfaz debe ser cercana al usuario para que se sienta cómodo al utilizarla. En mWallet se permite que el usuario configure una imagen de perfil.

Otro aspecto importante que debería cumplir una aplicación mobile wallet es que los pagos se realicen de forma rápida, no pudiendo tardar más que uno tradicional. Por tanto, el número de pasos necesarios para realizar una tarea debe ser el mínimo posible.

Algunos usuarios mencionaron que sería de gran utilidad una sección con un breve manual de ayuda.

A continuación se muestran un resumen de las características que debería tener un mobile wallet:

1. Completar automáticamente campos que sean propensos a cumplimentarse siempre de la misma manera en los formularios. Por ejemplo, la dirección de facturación que se indica cada vez que se añade una nueva tarjeta suele coincidir con las demás.
2. Utilizar la imagen y logotipo de cada establecimiento, permite al usuario identificarlos de una manera visual, lo que agiliza el uso de la aplicación. Este aspecto fue valorado por la mayoría de los usuarios.

3. Incluir un mapa si existe una sección que muestra los establecimientos cercanos que permiten realizar el pago con el mobile wallet. Este mapa debe poder ser ampliado para facilitar su consulta.
4. Diseñar una interfaz amigable y cercana al usuario, permitiendo personalizar algunos aspectos como por ejemplo incluir una imagen de perfil.
5. La fotografía también supone una medida de seguridad de cara a la identificación de la persona que paga en el establecimiento.
6. Navegación rápida y con el menor número de pasos para realizar cualquier funcionalidad, en especial los pagos en sus distintas modalidades.
7. Diseñar la aplicación de manera que transmita seguridad y confianza al usuario.
8. Incluir una sección de ayuda en la aplicación o bien un “Virtual Tour” al iniciarla por primera vez.
9. El uso de Cash Desk y Check In se puede implementar en una misma aplicación sin que ello suponga confusión a los usuarios.

6. Gestión del proyecto

La gestión de proyecto permite realizar una planificación y un seguimiento de las actividades que se llevan a cabo, así como estimar los recursos humanos y materiales que intervienen en el desarrollo.

Esto permitirá llevar un control sobre el estado del proyecto, pudiendo detectar problemas y solucionarlos rápidamente para minimizar el impacto sobre el tiempo de desarrollo y los costes.

6.1 Ciclo de vida del desarrollo de software

En el desarrollo de este proyecto se ha seguido un **ciclo de vida en cascada**, cuyas tareas se ajustan al proceso seguido durante el mismo.

El ciclo de vida en cascada (figura 32) define las siguientes fases:

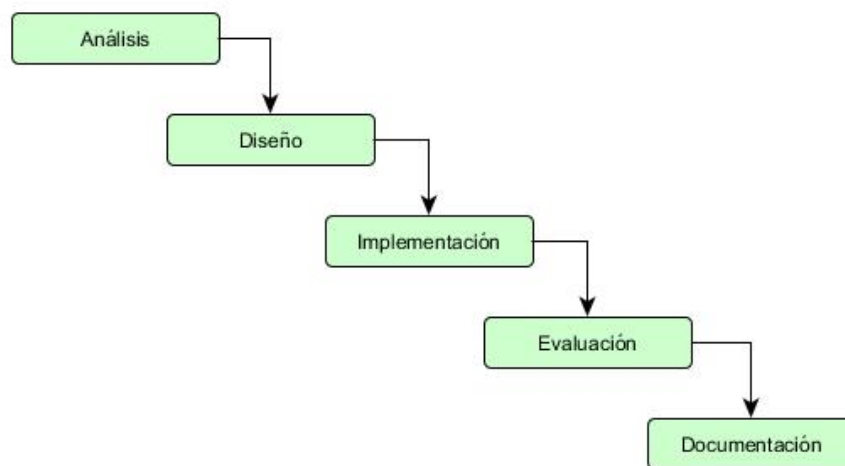


Figura 32. Ciclo de vida en cascada

- Análisis
Se estudian las necesidades de los usuarios, el estado del arte y se definen los requisitos que deberá tener el sistema.
- Diseño
Se define la arquitectura del sistema y los componentes del mismo, especificando las tareas de cada uno y las relaciones entre ellos.
- Implementación
Se desarrolla el sistema a partir de lo definido en las etapas anteriores.

- Evaluación
Se realizan pruebas sobre el sistema desarrollado con el fin de obtener un feedback de los usuarios y el cliente.
- Documentación

6.2 Planificación

El objetivo de la planificación es establecer fechas previstas para la realización de las tareas que componen el desarrollo del sistema. Véase tabla 62.

En primer lugar se deben definir las actividades y tareas que son necesarias y posteriormente se establecen los plazos de tiempo de cada una de ellas.

Tarea	Fecha de Inicio	Fecha de Fin	Duración (días)
Análisis	20-ene	03-feb	15
Estudio de aplicaciones similares	20-ene	23-ene	4
Definición de escenarios	25-ene	27-ene	3
Análisis de requisitos	28-ene	03-feb	7
Diseño	05-feb	22-feb	18
Estudio de las tecnologías	05-feb	06-feb	2
Diseño de la arquitectura	06-feb	09-feb	4
Diseño de la base de datos	09-feb	12-feb	4
Diseño de la interfaz de la aplicación	17-feb	22-feb	6
Implementación	27-feb	10-may	73
Scripts Base de Datos	27-feb	04-mar	6
Creación pantallas de registro	05-mar	07-mar	3
Creación pantalla principal	07-mar	13-mar	7
Creación sección Cartera	11-mar	31-mar	21
Creación formulario de nueva tarjeta	21-mar	24-mar	4
Creación sección Explora	18-abr	30-abr	13
Creación sección Actividad Reciente	04-abr	07-abr	4
Creación pantalla de Recibo	29-mar	01-abr	4
Creación sección Ofertas y Descuentos	08-abr	12-abr	5
Creación sección Comparte los gastos	01-may	07-may	7
Creación sección Configuración	25-mar	28-mar	4
Servicio Web "PerfilUsuario"	17-mar	24-mar	8
Servicio Web "GestionTarjetas"	18-mar	24-mar	7
Servicio Web "GestionPagos"	18-mar	25-mar	8
Desarrollo aplicación Check In establecimientos	08-may	10-may	3
Evaluación	25-may	01-jun	8
Diseño del test	25-may	26-may	2
Realización de las pruebas	29-may	01-jun	4
Total Proyecto(Días)	114		

Tabla 62. Planificación del proyecto

La programación de tareas para este proyecto se realiza a través de un diagrama de Gantt (figura 33).



Figura 33. Diagrama de Gantt de la planificación del proyecto

6.3 Presupuesto

En este apartado se elabora el presupuesto del proyecto, analizando los recursos, tanto humanos como materiales, necesarios para poder llevar a cabo las tareas detalladas en el apartado anterior.

6.3.1 Recursos humanos

El coste en recursos humanos se realiza tomando en cuenta las siguientes consideraciones:

- La fecha de inicio del proyecto es el *20 de enero de 2014* y la fecha de finalización del mismo es el *1 de junio de 2014*, lo que resulta en un total de *114 días* de trabajo.
- Se realiza una dedicación de 4 horas diarias lo que hacen un total de *456 horas* de trabajo distribuidas en 5 meses.
- El proyecto ha sido desarrollado por una única persona que tomará los roles de los distintos recursos humanos que se detallan a continuación.

El personal necesario para la realización del proyecto estará relacionado con el ámbito del HCI (*Human-Computer Interaction*) y la User Experience [33].

- Diseñador de User Experience
Dirige el desarrollo y la comunicación del proceso de diseño centrado en el usuario. Participa en las fases de investigación, análisis, desarrollo y pruebas.
- Diseñador de la interacción
Crea nuevos conceptos de interacción que sean amigables al usuario y permitan realizar las tareas de forma simple y que atraigan a los usuarios.
- Arquitecto de la información
A partir de una definición de escenarios se encarga de diseñar wireframes, prototipos, especificaciones funcionales y otros artefactos que permiten describir la experiencia de usuario.
- Diseñador gráfico
A partir de los wireframes y mockups crea diseños atractivos, usables y efectivos.
- Analista de usabilidad
Garantiza que el sistema cumple los requisitos de usabilidad necesarios a través de evaluación y pruebas.
- Programador
Se encarga del desarrollo de la aplicación.

A continuación, en la tabla 63, se muestra el desglose de los costes debidos a personal que serían necesarios:

Categoría	Coste (€/hora)	Horas dedicadas	Total (€)
Diseñador de la interacción	44	60	2.640
Diseñador de <i>User Experience</i>	44	232	10.208
Arquitecto de la información	40	72	2.880
Programador	18	292	5.256
Analista de usabilidad	15	62	930
Diseñador gráfico	15	72	1.080
Total			22.994

Tabla 63. Costes de personal

6.3.2 Recursos materiales

El coste asociado a los recursos materiales usados para la realización del proyecto se muestra en la tabla 64.

Descripción	Coste (€)	Unidades	% de uso dedicado	Periodo de depreciación (meses)	Uso (meses)	Total (€)
Asus Zenbook UX32V	1000	6	100	24	5	208,33 (x6)
LG Nexus 4	350	3	100	18	5	97,22 (x3)
Material fungible	550	-	-	-	-	550
Total						2.091,61

Tabla 64. Costes materiales

La fórmula de la amortización utilizada es la siguiente:

$$\frac{A}{B} * C * D$$

Siendo:

- A: Número de meses que es utilizado el equipo
- B: Periodo de depreciación
- C: Coste del equipo (sin IVA)
- D: % de uso que se dedica al proyecto

6.3.3 Resumen de costes

Teniendo en cuenta los costes indicados en los apartados anteriores, el coste final del proyecto se muestra en la tabla 65.

Descripción	Coste (€)
Recursos Humanos	22.994
Recursos Materiales	2.091,61
Total	25.085,61

Tabla 65. Resumen de costes

El presupuesto total de este proyecto asciende a la cantidad de **VEINTICINCO MIL OCHENTA Y CINCO CON SESENTA Y UN EUROS**, al cual se le aplicará el IVA en el momento de la facturación.

Leganés a 15 de Junio de 2014

El ingeniero proyectista



Fdo. Carlos Alcalá García

7. Conclusiones y líneas futuras de desarrollo

7.1 Conclusiones

Los avances tecnológicos han hecho posible que dispongamos en la actualidad de numerosas funcionalidades en nuestros teléfonos y dispositivos móviles. Especialmente, podemos estar conectados casi en cualquier momento y en cualquier lugar. Esto permite desarrollar entornos ubicuos, de tal manera que los usuarios realizan sus tareas habituales despreocupándose del dispositivo que estén usando y centrándose únicamente en la tarea en sí.

Una de las acciones más habituales de nuestro día a día es pagar. Las aplicaciones mobile wallets están creciendo cada vez más y en los próximos años, gran parte de los pagos y transacciones que se realicen se harán por medio de dispositivos móviles, principalmente smartphones.

Ya existen aplicaciones que se utilizan a diario con relativo éxito. Sin embargo, todas ellas tienen limitaciones que frenan su expansión. Así, por ejemplo, Google Wallet está disponible solo en Estados Unidos y solo utiliza la tecnología NFC; la aplicación móvil de PayPal no es operativa en su totalidad en España; Isis está vinculada a las operadoras AT&T Mobility, T-Mobile USA y Verizon, no pudiendo ser utilizada por clientes de otras compañías.

Pero el mayor límite que impide que estas aplicaciones puedan ser usadas de manera universal es que proporcionan modalidades distintas de interacción, impidiendo que los usuarios puedan aprovechar la *user experience* y tengan que aprender de nuevo cada aplicación.

Es por tanto necesario desarrollar un sistema de pago mediante móvil que pueda ser adoptado por una gran mayoría de usuarios, establecimientos y entidades financieras.

La **contribución** de este proyecto al conocimiento científico no se limita al desarrollo de una aplicación móvil. De hecho, en primer lugar he definido formalmente dos modelos de interacción orientados a crear un estándar para realizar los pagos mediante una cartera electrónica (Cash Desk y Check In). Tras realizar una prueba con usuarios se han extraído nueve lecciones (*Lecciones Aprendidas*) que pueden ser utilizadas por otros diseñadores como punto de partida para elaborar aplicaciones similares.

Además, al coexistir en la aplicación ambas modalidades de pago y al ser posible realizarlas utilizando las distintas tecnologías existentes (NFC, QR y geolocalización) se amplía el abanico de dispositivos y establecimientos que pueden utilizarla.

El medio empleado para conseguir hacer efectivas esas dos modalidades de pago citadas ha sido la elaboración de un prototipo de aplicación siguiendo la metodología Design Science Research (DSR) y obteniendo nuevos conocimientos a partir de éste.



Tras elaborar unos escenarios teóricos donde pueden ser utilizadas las distintas modalidades de pago y extraer a partir de ellos los requisitos que debe cumplir mWallet, se ha procedido a diseñar la aplicación.

Este prototipo se ha desarrollado basándose en los conceptos de la interacción persona - ordenador (*Human-Computer Interaction*) con la finalidad de minimizar errores, reducir la curva de aprendizaje y conseguir un uso eficiente del mismo.

Para ello se ha desarrollado una interfaz sencilla y homogénea para ambas modalidades de pago.

Igualmente, se ha buscado conseguir un entorno amigable al usuario, permitiendo incluir una imagen de perfil, que además proporciona una seguridad añadida a los pagos mediante Check In.

Con el fin de facilitar la navegación a través de la aplicación se han seguido las guías de diseño de Android [24], aprovechando así la *user experience* de los usuarios de este sistema operativo.

También se han simplificado las tareas que deben realizar los usuarios de la aplicación mWallet, utilizando para ello selectores de fechas, logotipos de los establecimientos, mapas que permiten ver la ubicación del usuario y los comercios cercanos. Todo esto además permite que los errores que se puedan cometer sean mínimos.

Una vez realizado el prototipo, se ha procedido a evaluarlo con diferentes personas que han aprendido la aplicación, la han utilizado y han expresado su opinión sobre la misma.

El resultado de esta evaluación ha demostrado que el diseño realizado, tanto para la interfaz como para el flujo de trabajo de las modalidades de pago es sencillo de utilizar, efectivo y rápido. Una vez aprendida la aplicación el usuario recuerda fácilmente como se realizan las tareas y el tiempo que emplea en realizar un pago es muy corto.

La mayoría de los usuarios consultados utilizarían como instrumento de pago la aplicación mWallet y elegirían alguna de las dos modalidades de pago desarrolladas en función de los escenarios en que se encontrasen.

Por supuesto, se trata de una evaluación muy limitada (test piloto). Quedaría pendiente completar la arquitectura final, dotarla de estrictas medidas de seguridad y someter a la aplicación a una prueba final con un mayor número de usuarios y en entornos reales.

El prototipo de aplicación mobile wallet desarrollado se ha implementado con una arquitectura cliente-servidor que ha obligado a desarrollar no solo la aplicación móvil sino también una base de datos MySQL relacional y un servidor web tomcat. Utilizando todas estas herramientas se ha podido comprobar el grado de cumplimiento del objetivo principal de este proyecto, consistente en el desarrollo de dos modalidades de pago, Cash Desk y Check In, que se han puesto de manifiesto como útiles y eficaces.

7.2 Líneas futuras de desarrollo

Como se vio en la introducción, una aplicación mobile wallet con muchas funcionalidades sería difícil de abarcar y podría provocar que los usuarios no la utilizaran por su dificultad.

Por ello, las líneas futuras de desarrollo deberían limitarse a facilitar las tareas cotidianas más habituales, que no complicasen en exceso el uso de la aplicación. Se trata de centrarse en la tarea olvidándose del medio que se está utilizando para realizarla.

Las principales líneas de desarrollo futuro serían:

- Elevar el prototipo desarrollado a la categoría de aplicación definitiva implementando una arquitectura completa y medidas de seguridad en colaboración con establecimientos comerciales y entidades financieras.
- Ampliar el área de uso de la aplicación incorporando a la misma tarjetas de transporte público, tarjetas de embarque y tarjetas sanitarias.
- Dotar a la aplicación de funcionalidades añadidas que puedan ser de utilidad para el usuario: estadísticas de gastos, posibilidad de reserva de mesa al hacer Check In en restaurantes y similares.
- Completar las pruebas en un entorno real y con más usuarios para comprobar las lecciones aprendidas en este proyecto y sacar guías de diseño generales que sean efectivas y útiles para los diseñadores de mobile wallets.
- Desarrollar la aplicación para otros sistemas operativos (iOS, Windows Phone...)

Bibliografía

- [1] Merchant Warehouse (2014), The Mobile Wallet. Consultado en Junio de 2014. <http://merchantwarehouse.com/state-of-the-payments-industry/mobile-wallet>
- [2] Merchant Warehouse (2013), The Mobile Payments Revolution. Consultado en Junio de 2014. http://merchantwarehouse.com/sites/merchantwarehouse.com/files/Mobile_Payments_Revolution.pdf
- [3] Business Insider (2014), Starbucks Generated Over \$1 Billion From Mobile Transactions Last Year. Consultado en Junio de 2014. <http://www.businessinsider.com/mobile-payments-at-starbucks-explode-in-2013-passing-the-1-billion-mark-2-2014-1>
- [4] Expansión (2013), Vodafone lanza en España el primer servicio de pago por móvil en Europa. Consultado en Junio 2014. <http://www.expansion.com/2013/11/22/empresas/tmt/1385118969.html?a=44a4196bc0ccb16f5b20d83600bf1605&t=1402639572>
- [5] Karnouskos, Stamatis, et al. "Guest Editorial: Research advances for the mobile payments arena." *Electronic Commerce Research and Applications* 7.2 (2008). P.137-140.
- [6] Stamatis Karnouskos and András Vilmos. 2004. "The European Perspective on Mobile Payments". *IEEE Symposium on Trends in Communications (SympoTIC '04), 24-26 October 2004, Bratislava, Slovakia*, p. 195 - 198
- [7] Mia Olsen, Jonas Hedman, and Ravi Vatrapu. 2012. "Designing digital payment artifacts". *Proceedings of the 14th Annual International Conference on Electronic Commerce (ICEC '12)*, p. 161 - 168
- [8] Tim McLaren, M. B. A., and Paul Buijs. "A design science approach for developing information systems research instruments".
- [9] Flood, Darren, Tim West, and Daniel Wheadon. "Trends in Mobile Payments in Developing and Advanced Economies." *RBA Bulletin* (2013), p. 71-80.



[10] El País (2013), BBVA lanza una aplicación de pagos con móvil. Consultado en Mayo de 2014 http://tecnologia.elpais.com/tecnologia/2013/12/03/actualidad/1386067908_522338.html

[11] Google Wallet (2014). Consultado en Mayo de 2014 <https://www.google.com/wallet/>

[12] PayPal (2014). Consultado en Mayo de 2014 <https://www.paypal.com/es/webapps/mpp/home>

[13] BBVA (2014). BBVA Wallet. Consultado en Mayo de 2014 <https://www.bbva.es/estaticos/micros/wallet/>

[14] Isis (2014). Pay with your phone. Consultado en Mayo de 2014. <https://www.paywithisis.com/>

[15] Bemoov (2014). Consultado en Junio de 2014. <http://www.bemoov.it/>

[16] PingPing (2014). Consultado en Junio de 2014. <http://www.pingping.be/wp/>

[17] Quick Tap (2014). Consultado en Mayo de 2014. <https://register.quicktap.com.au/Home.aspx>

[18] Rogers, Yvonne, Helen Sharp, and Jenny Preece. Interaction design: beyond human-computer interaction. John Wiley & Sons, 2011

[19] Technovelgy. How RFID Works. Consultado en Mayo de 2014. <http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=2>

[20] TransTech Systems (2012). Passive vs Active Tags. Consultado en Mayo de 2014 <http://assettrackit.com/passive-vs-active-tags/>

[21] Coskun, Vedat, Busra Ozdenizci, and Kerem Ok. "A Survey on Near Field Communication (NFC) Technology." *Wireless personal communications* 71.3 (2013); p. 2259-2294.

[22] KIESEBERG, Peter, et al. QR code security. En Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia. ACM, 2010. p. 430-435.

[23] NotixTech (2011). 9 QR Code Data Types. Consultado en Mayo de 2014. <http://notixtech.com/blog/9-qr-code-data-types>

[24] Android Developers (2014). Design. Consultado en Marzo de 2014 <http://developer.android.com/design/index.html>

[25] Android Developers (2014). Fragments. Consultado en Junio de 2014. <http://developer.android.com/guide/components/fragments.html>

[26] Android Developers (2014). Storage Options. Consultado en Junio de 2014. <http://developer.android.com/guide/topics/data/data-storage.html>

[27] w3Schools (2014). SOAP. Consultado en Junio de 2014. http://www.w3schools.com/webservices/ws_soap_intro.asp

[28] Sgoliver.net (2012). Tareas en segundo plano en Android (I): Thread y AsyncTask. Consultado en Mayo de 2014. <http://www.sgoliver.net/blog/?p=3099>

[29] Sgoliver.net (2012). Notificaciones Push Android: Google Cloud Messaging (GCM). Introducción. Consultado en Mayo de 2014. <http://www.sgoliver.net/blog/?p=2682>

[30] Android Developers (2014). Google Maps Android API v2. Consultado en Junio de 2014. <http://developer.android.com/google/play-services/maps.html>

[31] ZXing (2014). Official ZXing ("Zebra Crossing") project home. Consultado en Mayo de 2014 <https://github.com/zxing/zxing>

[32] STC Usability & User Experience Community (2001). Measuring Usability with the USE Questionnaire. Consultado en Junio de 2014. http://www.stcsi.org/usability/newsletter/0110_measuring_with_use.html

[33] Onward Search (2011). A Guide to UX Careers. Consultado en Junio de 2014.

<http://www.onwardsearch.com/UX-Career-Guide/>

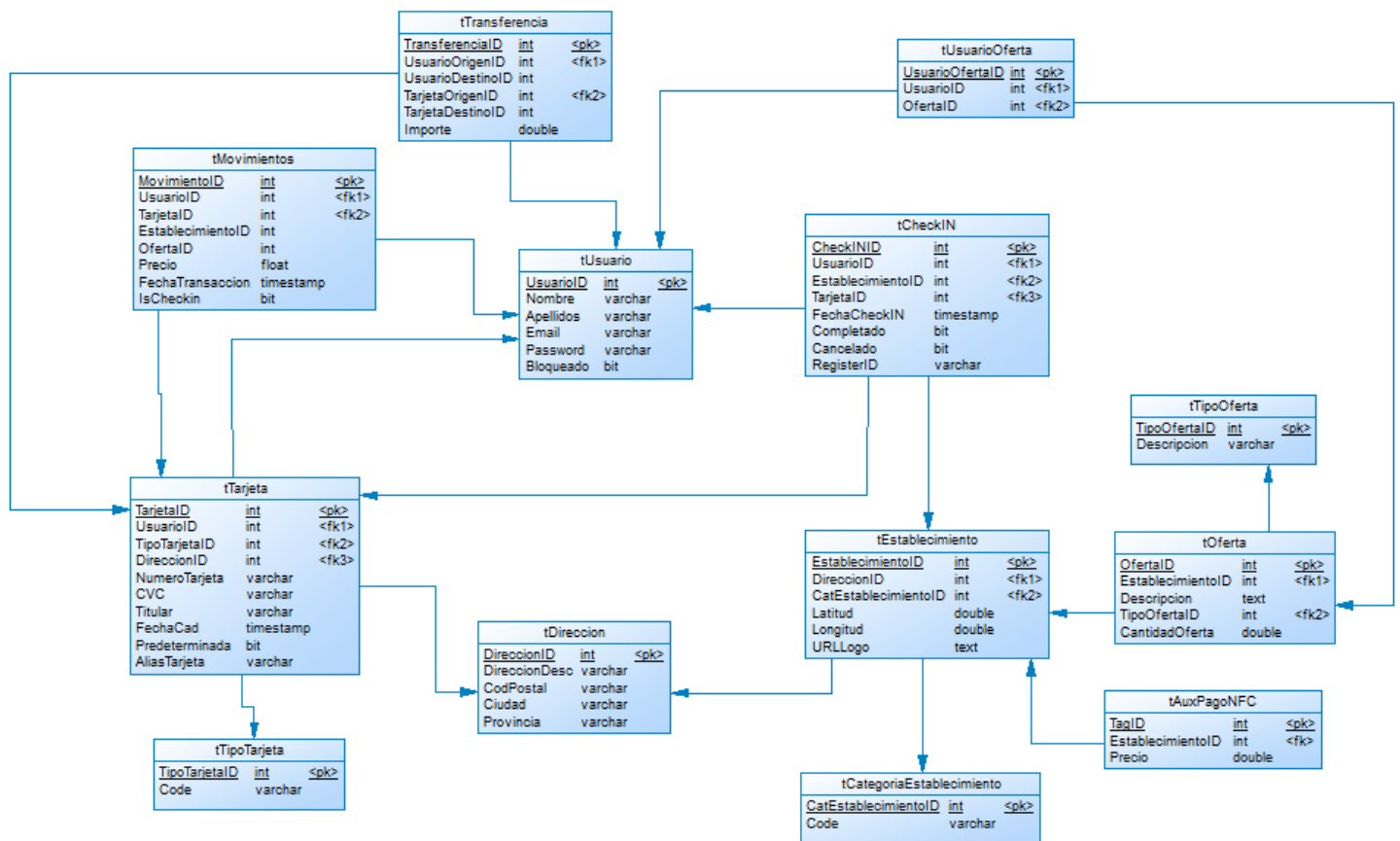
[34] Payment Card Industry, Requirements and Security Assessment Procedures, Version 3.0 (Noviembre 2013)

[35] Romano, Marco. *Designing Usable Mobile Interfaces for Spatial Data*. Diss. Doctoral Thesis. University of Salerno, 2012.

[36] Android Developers (2014). Navigation Drawer. Consultado en Junio de 2014. <https://developer.android.com/design/patterns/navigation-drawer.html>

[37] Nielsen Norman Group (1995). 10 Usability Heuristics for User Interface Design. Consultado en Junio de 2014. <http://www.nngroup.com/articles/ten-usability-heuristics/>

Anexo I. Diseño de la base de datos



TABLAS PRINCIPALES

tUsuario

Esta tabla almacena toda la información relativa a un usuario registrado en mWallet: nombre, apellidos, email y contraseña de acceso.

Además, se incluye un campo “Bloqueado” que especifica si la aplicación se encuentra o no bloqueada para el usuario. Este campo se incluye por motivos de seguridad, ya que el usuario, en caso de pérdida o robo podrá bloquear la aplicación evitando así posibles fraudes.

tTarjeta

Guarda toda la información de las tarjetas de los usuarios de la aplicación: número, CVC, titular y fecha de caducidad.

Junto con los datos básicos también se incluyen campos de control “Predeterminada” que servirá a la aplicación para conocer cuál es la tarjeta predeterminada que debe mostrar al usuario cuando va a seleccionar una tarjeta.

Una tarjeta está relacionada con un usuario (tUsuario), una dirección de facturación (tDireccion) y un tipo de tarjeta (tTipoTarjeta). Estas relaciones se establecen mediante foreign keys con los identificadores de cada una de las tablas con las que está relacionada tTarjeta.

También se guarda un alias de la tarjeta que sigue el formato: TipoTarjeta – x-dddd, siendo “dddd” los últimos cuatro dígitos de la tarjeta de crédito, por ejemplo: Visa x-1234.

Este alias será el que se enviará a la aplicación cuando se soliciten los datos de las tarjetas del usuario, evitando así tener que enviar el número de la tarjeta completo.

tDireccion

Esta tabla contiene los datos relativos a las direcciones, tanto de facturación de las tarjetas de crédito, como de los establecimientos que permiten el pago con mWallet.

La dirección incluye la descripción (nombre de la vía), el código postal, ciudad y provincia.

tEstablecimiento

Almacena la información relativa a los establecimientos que permiten realizar los pagos con mWallet.

Junto con los datos básicos del establecimiento se almacenan también la latitud y longitud, con estos datos, el servidor puede calcular la distancia a un establecimiento desde la posición del usuario mediante la fórmula de Haversine.

También se guarda la URL de la imagen del establecimiento para que pueda ser cargada por la aplicación en tiempo de ejecución.

tMovimientos

Contiene los datos de las transacciones realizadas por los usuarios de mWallet. Por cada transacción se almacenan los identificadores de usuario, tarjeta, establecimiento y oferta en el caso de que el usuario haya utilizado alguna.

Además, es necesario distinguir si es un pago completado o es un check in activo, para ello se ha incluido un campo binario “IsCheckin” que permite saber a la aplicación si se trata de un check in o no.

tCheckIN

Además de almacenar en la tabla tMovimientos un campo que indica si es check in o no también se utiliza una tabla específica para la gestión de este tipo de transacciones.

La tabla, además de los datos de usuario, establecimiento y tarjeta también contiene campos de control, estos son: Completado y Cancelado. Ambos son campos binarios que permiten a los establecimientos gestionar los check in activos.

En el momento en el que un check in es completado, el dispositivo móvil debe recibir una notificación indicando que el pago se ha realizado correctamente. Para esta comunicación iniciada por el servidor se ha utilizado Google Cloud Messaging, que será explicado en detalle en los apartados 4.2.2. Este servicio necesita un identificador del dispositivo para poder enviar el mensaje al mismo, este identificador es almacenado en el campo RegisterID de esta tabla.

tTransferencia

La tabla contiene la información de los usuarios y tarjetas entre los que se realiza la transferencia así como el importe de la misma.

tOferta

Esta tabla almacena la información de las ofertas ofrecidas por los establecimientos. Para cada oferta se tiene la descripción, el establecimiento que la ofrece y el tipo de oferta.

Para el prototipo diseñado se han considerado 3 tipos de oferta:

- Descuento: Se reduce el precio de la compra en una cantidad.
- Porcentaje: Se realiza un descuento calculado a partir de un porcentaje sobre el importe de la compra.
- lxC: Son ofertas del tipo 2x1

tUsuarioOferta

Una oferta puede estar asociada a muchos usuarios y un usuario puede tener muchas ofertas, por eso se ha incluido la tabla tUsuarioOferta.

La tabla asocia una oferta a un usuario. Se almacenan los identificadores de usuario y oferta.

TABLAS DE CÓDIGOS

Las tablas de código solo tienen 2 campos, un identificador, que será el utilizado por las tablas principales para realizar las referencias y un valor asociado a ese identificador.

tTipoTarjeta

Almacena los distintos tipos de tarjeta soportados en mWallet: Visa, Mastercard ...

tTipoOferta

Contiene la información sobre los distintos tipos de oferta que se pueden aplicar en las compras.

Estos pueden ser de tres tipos:

- Descuento expresado en cantidad monetaria
- Descuento expresado en porcentaje
- Descuento expresado en unidades o cantidades gratis (por ejemplo: 2x1)

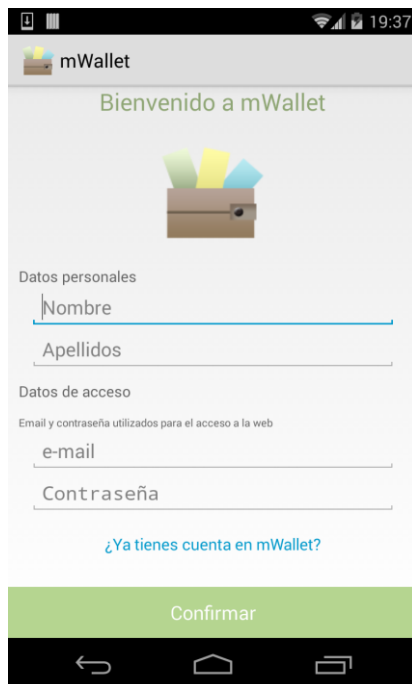
tCategoriaEstablecimiento

Almacena las distintas categorías que puede tener un establecimiento: deportiva, restauración, moda ...

Anexo II. Encuesta de evaluación


UTILIDAD						
1. Me permite ser más efectivo al realizar los pagos	1	2	3	4	5	NA
2. Me permite ser más productivo	1	2	3	4	5	NA
3. Es útil	1	2	3	4	5	NA
4. Me proporciona más control sobre mis actividades diarias	1	2	3	4	5	NA
5. Me permite ahorrar tiempo al usarla	1	2	3	4	5	NA
6. Cubre mis necesidades	1	2	3	4	5	NA
7. Hace todo lo que espero que haga	1	2	3	4	5	NA
FACILIDAD DE USO						
8. Es fácil de usar	1	2	3	4	5	NA
9. Es amigable al usuario	1	2	3	4	5	NA
10. Necesita pocos pasos para realizar la tarea	1	2	3	4	5	NA
11. Es flexible	1	2	3	4	5	NA
12. No es necesario realizar un gran esfuerzo para usarla	1	2	3	4	5	NA
13. Se puede usar sin instrucciones	1	2	3	4	5	NA
14. No noto inconsistencias al usarla	1	2	3	4	5	NA
15. Gustará tanto a usuarios frecuentes como ocasionales	1	2	3	4	5	NA
16. Puedo deshacer mis errores rápidamente	1	2	3	4	5	NA
17. La uso correctamente siempre	1	2	3	4	5	NA
FACILIDAD DE APRENDIZAJE						
18. He aprendido a usar mWallet rápidamente	1	2	3	4	5	NA
19. Recuerdo fácilmente cómo se usa	1	2	3	4	5	NA
20. Es fácil aprender a usarla	1	2	3	4	5	NA
21. He adquirido las habilidades necesarias para usarla fácilmente	1	2	3	4	5	NA
SATISFACCIÓN						
22. Estoy satisfecho con la aplicación	1	2	3	4	5	NA
23. Se la recomendaría a un amigo	1	2	3	4	5	NA
24. Es divertida de usar	1	2	3	4	5	NA
25. Funciona de la forma que quiero	1	2	3	4	5	NA
26. Es maravillosa	1	2	3	4	5	NA
27. Siento que necesito tenerla	1	2	3	4	5	NA
28. Es agradable	1	2	3	4	5	NA

Anexo III. Interfaz de mWallet



mWallet

Bienvenido a mWallet



Datos personales

Nombre

Apellidos

Datos de acceso

Email y contraseña utilizados para el acceso a la web

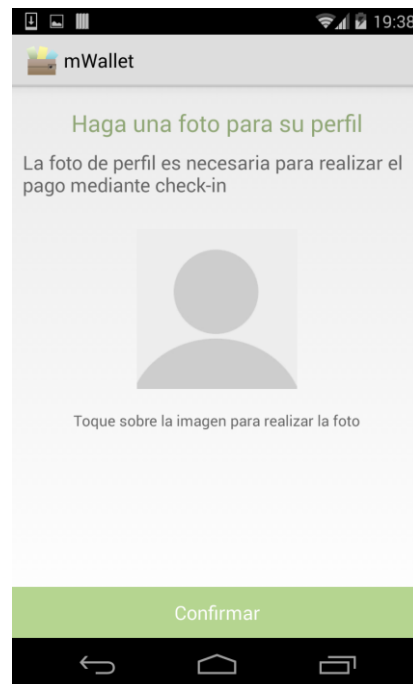
e-mail

Contraseña

[¿Ya tienes cuenta en mWallet?](#)

Confirmar


1. Formulario de registro



mWallet

Haga una foto para su perfil

La foto de perfil es necesaria para realizar el pago mediante check-in



Toque sobre la imagen para realizar la foto

Confirmar

2. Realizar foto de perfil



mWallet

Haga una foto para su perfil

La foto de perfil es necesaria para realizar el pago mediante check-in



Toque sobre la imagen para realizar la foto

Confirmar

3. Foto de perfil realizada



mWallet

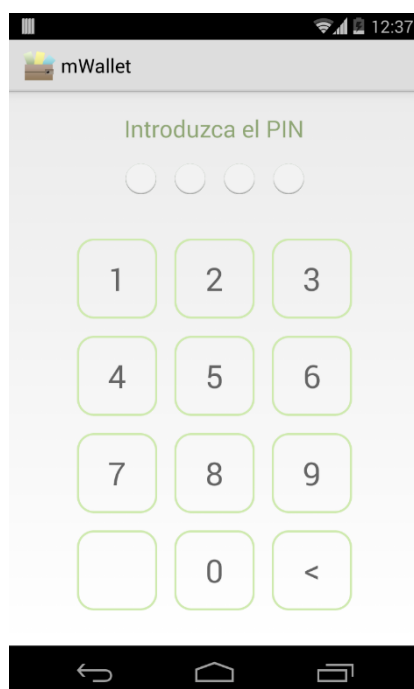
Configuración inicial completada

 Carlos
Alcalá García
mailto:mailtest@mail.com

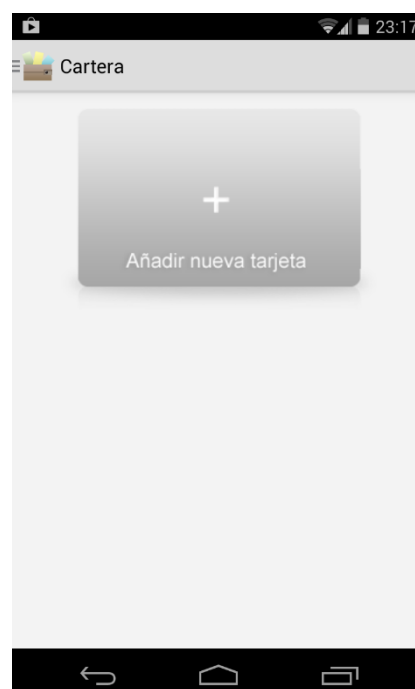
Puede modificar la información de su perfil en cualquier momento desde la pantalla de configuración.

Comenzar

4. Confirmación de registro

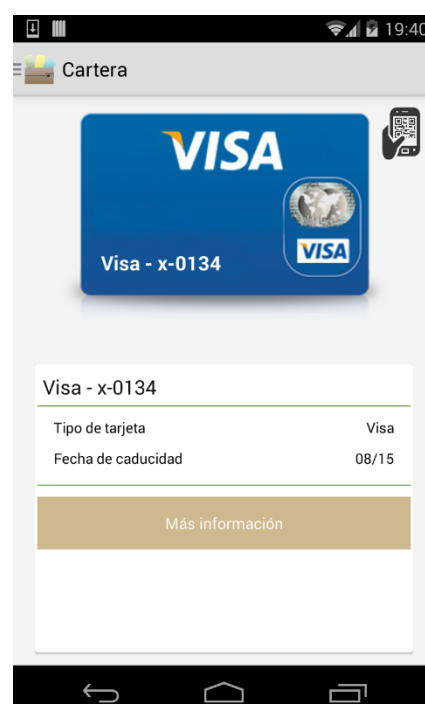


5. Introducir PIN

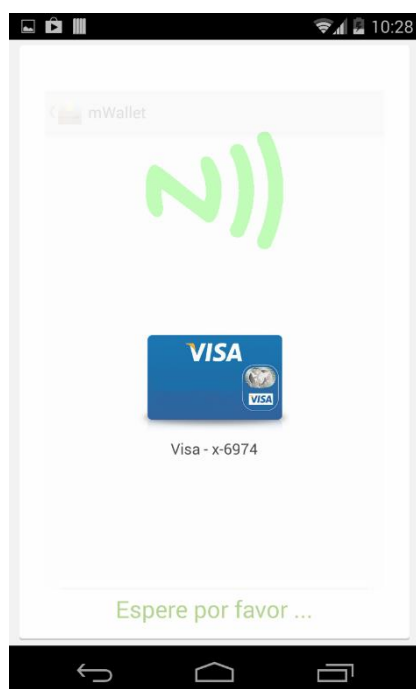


6. Pantalla primer uso cartera

7. Formulario nueva tarjeta



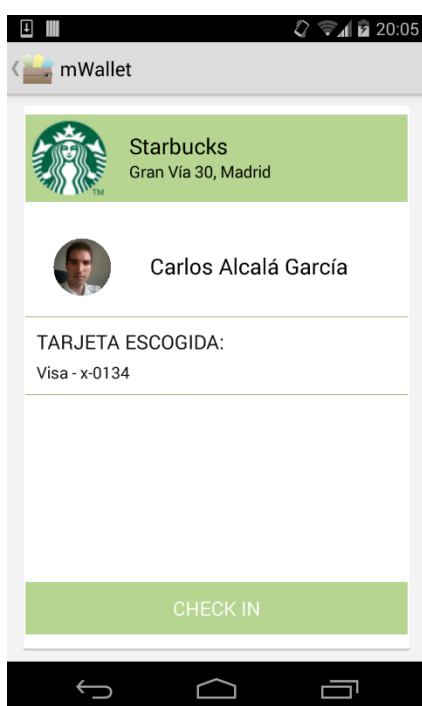
8. Cartera con tarjeta añadida



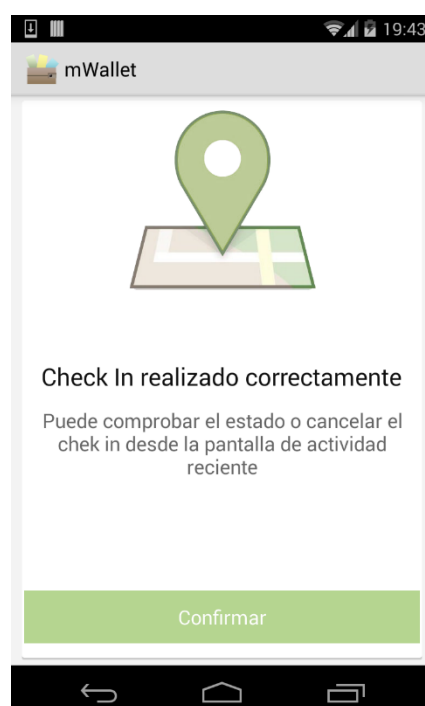
9. Cash Desk NFC (tarjeta reconocida)



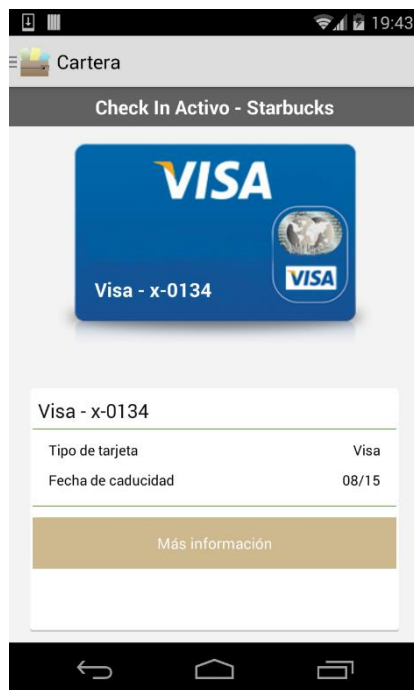
10. Recibo



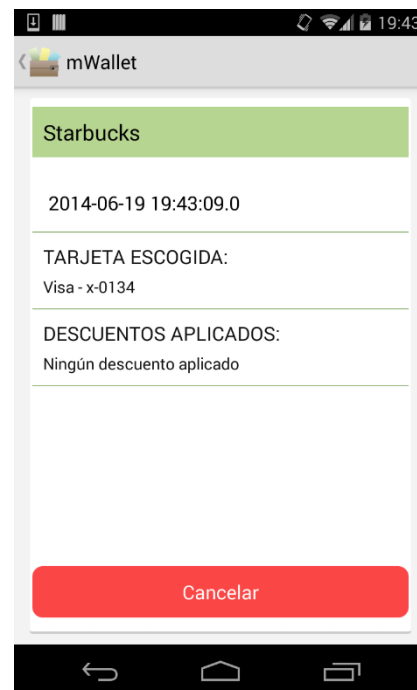
11. Check In mediante QR



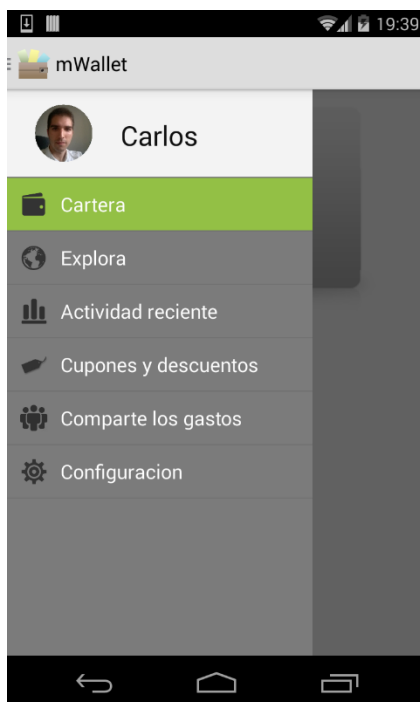
12. Confirmación Check In



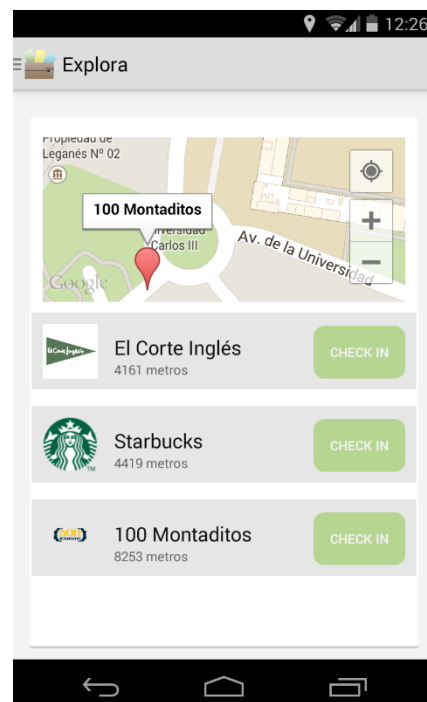
13. Aviso de Check In activo



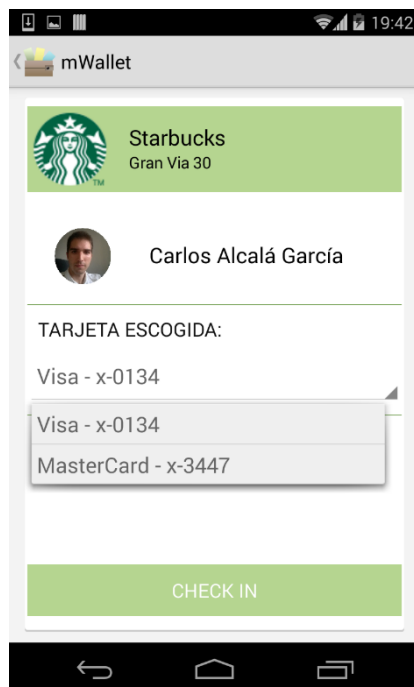
14. Cancelar Check In



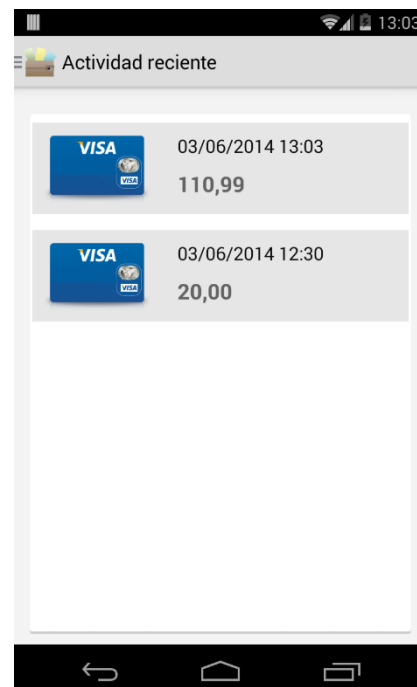
15. Menú lateral



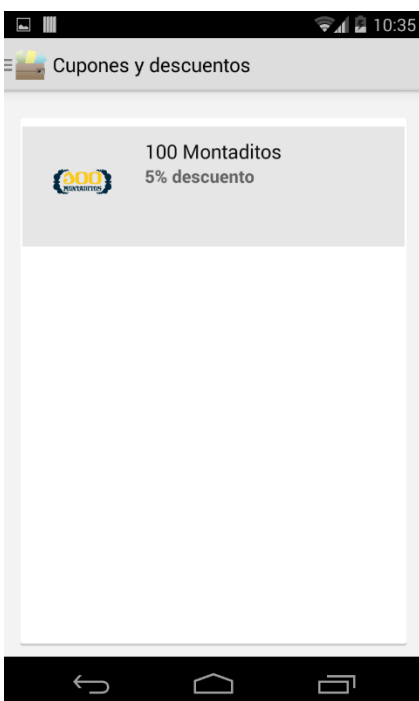
16. Sección explora



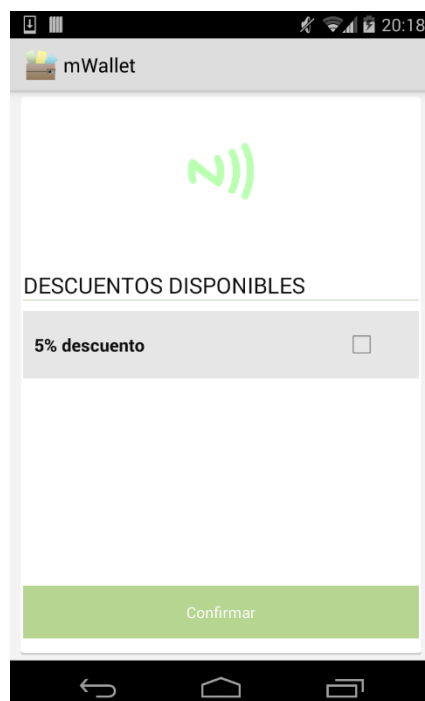
17. Check In con geolocalización



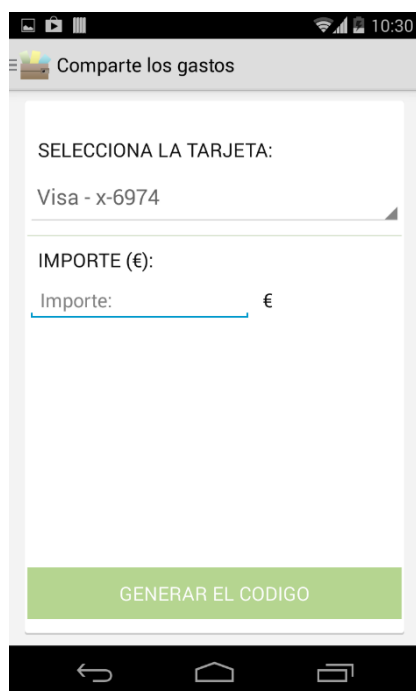
18. Sección Actividad Reciente



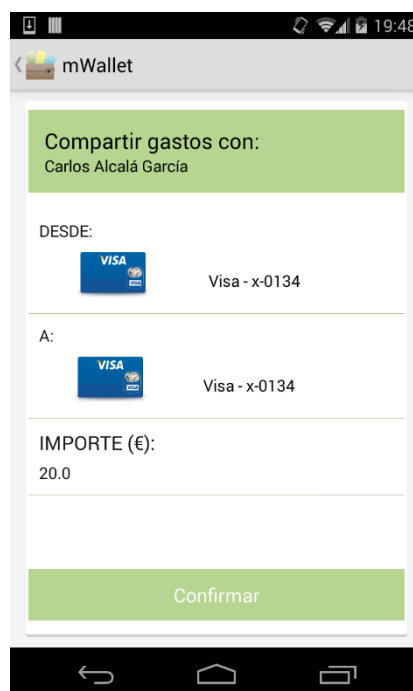
19. Sección Cupones y Descuentos



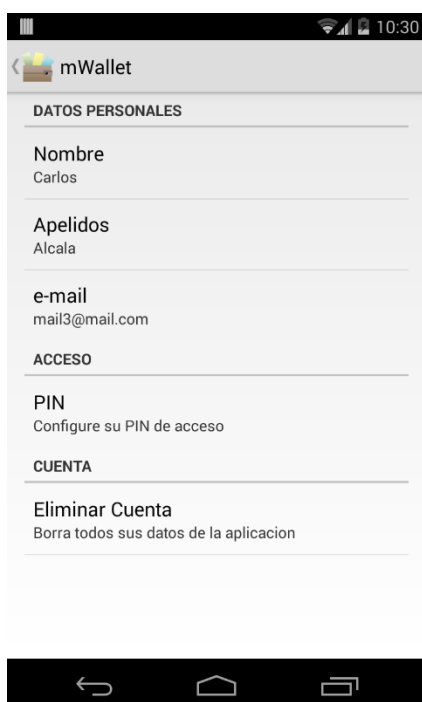
20. Aplicar descuentos al pago



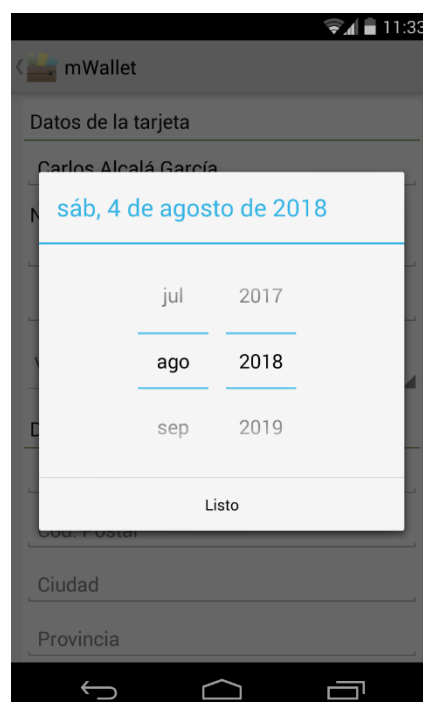
21. Comparte los gastos (generar QR)



22. Confirmar Compartir Gastos



23. Configuración



24. Selector de fecha de caducidad

Anexo IV - Project Summary

Abstract

The use of the smartphones have experienced a faster growth in the recent years. This devices include NFC readers, cameras, internet connection, and a lot of sensors which allow knowing, e.g., the user's location or the temperature. This enables the device to adapt to the environment and "learn" about the user's profile, letting the developers to create ubiquitous systems that reduce the daily tasks' effort.

Nowadays there are some apps that use the NFC technology and geolocation to make payments in stores using the smartphone. A lot of restaurants and other establishments are getting adapted to these new payment methods due to in the next years is being expected that a substantial amount of the payments will be made using the smartphone's apps.

However, current mobile wallet apps use different interaction to perform the purchases, this may lead the users to get confused. There is a lack of guidelines to design the payment interaction.

This bachelor thesis focuses on the definition of two interaction models and the development of a mobile application prototype aiming to overcome the current applications' limitations and demonstrate the efficacy and efficiency of these models.

In this project two new mobile payment interaction models have been designed: cash desk, where the interaction is similar to paying with a credit card, and check-in, where the consumer just has to register when he/she enters in the store, not having to do any other task to make the payment.

Keywords: Ubiquitous computing, Smartphone, Mobile Wallet, payment interaction models, Cash Desk, Check In

Introduction

Nowadays, mobile devices are equipped with the latest technologies that allow users to stay connected and receive information continuously and in real time. Mobile devices take advantage of those features, letting the developers to create ubiquitous environments where the users can perform their day-to-day tasks focusing on the goal and not having to be aware of the mobile device.

This project has been made inside the Human Computer Interaction (HCI) area, in particular the ubiquitous computing.

On June 4th of 2014, the website “Merchant Warehouse”, [1] [2] echoing a survey conducted by the Carlisle & Gallagher Consulting group about the usage of mobile phones to perform purchases, predicts that by 2017 half of today’s Smartphone users will be using mobile wallets as their preferred payment method.

During 2013, Starbucks processed over 1\$ billion from mobile transaction [3].

In Spain [4], contactless NFC technology is getting adopted very fast and more than 1,7 million transactions had been performed in the last month using this payment method. Financial entities like BBVA, Bankia or ING Direct are collaborating along with Visa in this adoption process.

What is a mobile wallet?

A mobile wallet is an electronic version of the wallet we all carry on our pockets with added benefit from all the advantages that this new method provides.

Most of these mobile wallets allow us to carry all of our credit cards inside the mobile device, we could apply coupons and offers and know the nearest commerce’s’ location where we could use the mobile wallet to perform a purchase. In addition, we can manage all the transactions we had made and see their receipt.

Mobile wallets should provide an easy to use and effective tool to perform the payments.

Context

Why, if these new applications provides too many advantages, has not been imposed as the preferred payment method yet?

In the article “Research advances for the mobile payments arena” [5], the authors explain we are in an era where the mobile devices are becoming more and more sophisticated with bigger computing capacity. This features make these applications to be able to support real-time payments in an effective and efficient way.

However, the existing mobile wallet solutions approaches are limited to a specific country or a specific operator, which are quite far from a universally solution.

Moreover, these applications are not sufficiently user friendly and presents a difficult learning curve.

In [5], authors specify that there is not an efficient collaboration between Banks, financial entities and the mobile wallet developers, which is an essential requirement for the success of mobile payments.

During last year many mobile payment solutions have been introduced and some of them have evolved from the prototype level to market. Unfortunately many of them had a low adoption rate.

There are several efforts in the European Union [6] in order to obtain a mobile payment standardization, and make possible to support the emerging mobile payment applications.

To do this, is necessary research, design and develop new mobile payment services and adjust the legal and regulatory framework to this new payment modality.

One example is SEMOPS (Secure Mobile Payment Service), which aims at developing an universal and standardized system that will be able to handle both national and international mobile payments.

SEMOPS focuses on security, privacy, user’s trust and the system flexibility.

There are some other works [7] that have researched in order to identify functional and design properties that mobile wallets should have.

After interviewing four different user groups, the obtained result was an application with a lot of functionalities, which, probably, would confuse the users and could lead them to a situation where they don’t want to use the mobile wallet.

It is necessary to make more studies and testing in real environments to discover what the users really want.

Socio-economic context

This bachelor thesis is part of a socio-economic context which, even though is marked by an important economic crisis, could be considered as positive for the purpose of mWallet.

One of the reasons is the great development of telecommunications and the widespread use of mobile devices. This make the apps market to be booming.

Moreover, the main objective of this application is implement on it new mobile payments interaction models that could be considered useful by the users, commerces and financial entities. This standardization could lead to a widespread of this payment model, which could be a revitalizing element to the economy.

Objective

This project aims to design and develop a smart wallet in order to improve the micropayments activities.

The goal is to overcome the limitations of the existing systems due to the heterogeneity which presents today's mobile wallets interfaces, using in one application most of the existing technologies in such a way that the range of devices which could use it and the number of potential users is increased.

The main objective is to develop interaction models which allow the user to perform the payments synthesizing the most common models used in today's smart wallets.

In order to achieve this, it has been designed, implemented and evaluated a prototype based on these new interaction models to verify its efficiency and efficacy.

Design and implementation methodology

The methodology used in this project is the "Design Science Approach"[8].

This methodology follows an iterative process that allows to evaluate and design IT artifacts focusing on the usefulness of them and the ease to learn it.

At the end of the process we can learn from the results obtained, "you build an artefact to learn from it" [35].

The process is divided in three cycles:

- *Relevance Cycle:* Grounded on the current environment, and taking into account the several components (people, organizational system, technical system and opportunities) we can obtain the artifact requirements.
- *Rigor Cycle:* Using previous knowledge from existing artifacts and similar researches we design our prototype. For instance, in this project it has been followed a user centered design; to show the credit cards information I have followed the Payment Card Industry Data Security Standard [34] (PCI DSS).
- *Design Cycle:* Using the extracted requirements and taking advantage from the previous knowledge we can design and elaborate the prototype.
The last step consist in perform an evaluation with potential users where we can obtain feedback.

This is an iterative process. The environment changes, new requirements and new related knowledge could appear. These changes would be reflected in the design.

Figure 1 represents the Design Science Research approach cycles.

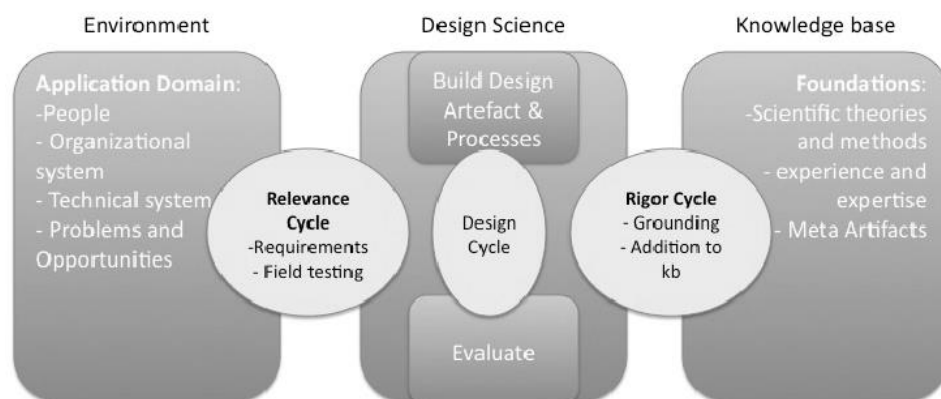


Figure 1. Design Science approach [35]

Outline

This document is structured in 7 chapters, one bibliography section and four annexes

The present chapter (*Chapter 1 - Introduction*) shows the mobile wallets current panorama. Although this applications provides a lot of advantages and the researches shows up that in the future this new payment methodology will be used by many users, it is still necessary to make efforts in order to found a standard which would make possible that this payment method could be used in a universal way.

After explaining the context I detail the objectives to be achieved in this project and the process followed to get them.

Chapter 2- State of the art, offers an overview and comparison of some of the most successful today's mobile wallets. Here it is make an analysis of them and see what technology they use and what interaction models they implement.

Chapter 3 - Interaction models and analysis, analyses the two designed interaction models, Cash Desk and Check In, which are implemented in mWallet.

This two models are explained through activity scenarios from real world where the two modalities are compared, revealing their advantages and disadvantages. From these scenarios are extracted the application requirements which are specified at the end of the chapter.

Chapter 4 - Design and implementation, details the mobile application design process (wireframes and mockup) and the workflow of the payment tasks. Moreover, it analyses the technologies that will be used to implement the two payment modalities (NFC and QR Codes). Finally it details the system architecture and the more important aspects related to the implementation.

Chapter 5 - Evaluation, shows the obtained results from the pilot test performed with ten potential users. This test aims to check the design usability, the efficacy and efficiency of the developed payment models.

Chapter 6 - Project management, describes the different project phases and the planning of the tasks of each one. After this it will be obtained the project budget, taking into account the human resources and the needed material to carry through the project.

Chapter 7 - Conclusions, summarizes the project and show up the future research lines which could be done to improve mWallet.

The document is completed with four annexes:

- The first one explain the database design. Here is detailed the different tables and the relations between them.
- The second annex shows the evaluation form template that has been used on the pilot test.
- Annex III - mWallet Interface, shows the different screenshots of the developed application.

- Annex IV - Project Summary, meet the requirement of translate into English the abstract and the introduction and conclusions chapters

Conclusions and Future Research

Conclusions

Technological advances in communications have made possible that nowadays we could have numerous functionalities on our phones and mobile devices. We could be almost connected every time, everywhere.

This features let the developers to create ubiquitous environments, so that the users perform their daily tasks without having to be aware of the device they are using, the user is completely focused on the task.

One of the most common tasks is to perform payments. Mobile wallets applications are growing more and more and in the following years is expected that many payments and transactions will be made using the mobile devices.

There are some applications which are being used nowadays with some success. However, all of them have some limitations that slow their expansion. For example, Google Wallet application is only available on the United States and only uses the NFC technology; PayPal mobile application is not completely operative in Spain; Isis is linked to AT&T Mobility, T-Mobile USA y Verizon, and can be used only if the user is a client of one of them.

The biggest limit that presents these applications and make very difficult to obtain an universal solution is that they provide different interaction modalities, making impossible to users to take advantage of the user experience. They have to learn every application. This can lead to a frustration on the user.

It is necessary to develop a mobile payment system which could be adopted by a great amount of users, commerces and financial institutions.

The **contribution** of this work to the scientific knowledge is not limited to the development of a mobile application. Indeed, first of all, I have formally defined two interaction models aimed at allowing a standard way to pay with a smart wallet. Then, I have formalized my experience in form of nine Learned Lessons that can be reused by other designers as starting point to deal with the design of similar applications

Moreover, due to both payment modalities (Cash desk and Check In) coexists in the same application, and the possibility to perform them using different technologies like NFC, QR Code or geolocation, the range of devices, users and establishments which could use it get increased.

In order to make effective those two payment interaction models I have developed a prototype application following the methodology Design Science Research (DSR). In this methodology the developer can obtain new knowledge and apply it to modify the designed artifact.

First of all I have described some theoretical scenarios (activity scenarios) where I show how the different payment modalities could be used. From these scenarios it is possible to extract the requirements that mWallet must satisfy.

Once the requirements were defined I started to develop the prototype.

This prototype has been developed based on the concepts of Human-Computer Interaction (HCI) aiming to minimize errors, reduce the learning curve and obtain an efficient use of the mobile wallet.

It has been developed a simple and homogeneous interface for both payment modalities in order to achieve this goal.

Equally, I have designed an user friendly interface, letting the user to establish a profile picture, which provides an extra security to the payments performed using the Check In interaction model.

With the purpose of make easy the navigation through the application I have followed the Android design guidelines [24], taking advantage from the user experience that this operating system's user have.

In addition, It has been also simplified the mWallet user's tasks, using date pickers, store's logotypes, maps which allow the user to see their position and the nearest establishments. This can lead to minimize the number of errors that the users can commit.

Once the prototype was created, I have made a pilot test with ten users which learnt the application, used it and expressed their opinion about mWallet.

The results obtained demonstrate that both, the interface design and the interaction models workflow is simple, useful, effective and fast. Once the user has learnt the application he can remember how the task are performed and the employed time to complete the task is very short.

Most of the users that participate in the pilot test would use the application mWallet as their preferred payment method and they would use some of the interaction models designed depending on the scenario.

This pilot test is very limited and it would be necessary to complete the final architecture, provide strict security measures to it and make a final evaluation in real environments (i.e. restaurant) with more users.

The mobile wallet application prototype I have developed is implemented using a client-server architecture so that I have to design along with the application a MySQL database and a tomcat webserver.

Using all these tools I had been able to check the accomplishment of the main goal of this project, consisting on the design of two payment modalities, Cash Desk and Check In, which have shown that are useful and efficient.

Future Research

As mentioned on the Chapter 1 - Introduction, a mobile wallet application which has a lot of functionalities would be difficult to use and learn, and probably confuse the users.

Future research lines should be focused on easing the day-to-day tasks making an application which would be friendly to the user.

The main future research lines would be:

- Elevate the developed prototype to a final application that could be used in the real market, implementing a complete architecture and strict security measures collaborating with the merchants and financial entities.
- Extend the use domain of the application including public transport cards, boarding passes and sanitary cards.
- Give the applications added functionalities that could be useful for the user like payment statistics, the possibility to book a table on a restaurant when the user uses the Check In modality.
- Complete the evaluation on a real environment and with more users in order to check the nine learned lessons in this project and extract general guidelines which could be effective and useful to the mobile wallets designers.

Summary

To conclude I will make a brief summary where I will explain chronologically the different phases I performed to elaborate this bachelor thesis.

I started making an overview and comparison of some of the most successful mobile wallets, reaching the conclusion that, although these applications have many advantages, they all have some limitations that make impossible to use them universally. It is necessary to make efforts to obtain a standard that could be used to develop these applications.

After, I elaborated three activity scenarios which explain different situations where mWallet can be used. Each scenario compares the two payment interaction models designed, Cash Desk and Check In. Both models allow the user to perform the payment using the technology they prefer (NFC, QR or Geolocation).

From these scenarios I extracted the requirements that mWallet must meet.

In order to evaluate the effectivity and efficacy of the designed interaction models I have developed, following the requirements defined in the previous phase, an android application.

I choose the android because it is one of the most widespread operating system in the mobile devices nowadays.

First of all, I designed the mobile interface based on the concepts of the HCI and in particular, the ubiquitous computing. The interface must be the same for each payment modality and to make them easy to learn I take advantage from the user experience.

Once the prototype is completed I performed a pilot test with ten users. From this evaluation I could extract some interesting results.

First of all I could see that the two payment models, Cash Desk and Check In, are useful, effective and easy to learn. Most of the users will use them to perform their payment.

Secondly, I could define nine learned lessons that must be taken into account to develop a mobile wallet.

Along with the application I designed a MySQL database and implemented three web services which communicate the application and the database. These web services are hosted on a tomcat 7.0 webserver.

The project I have presented in this document could be improved, adding new functionalities to it and making a complete architecture in collaboration with the financial entities and the merchants.

